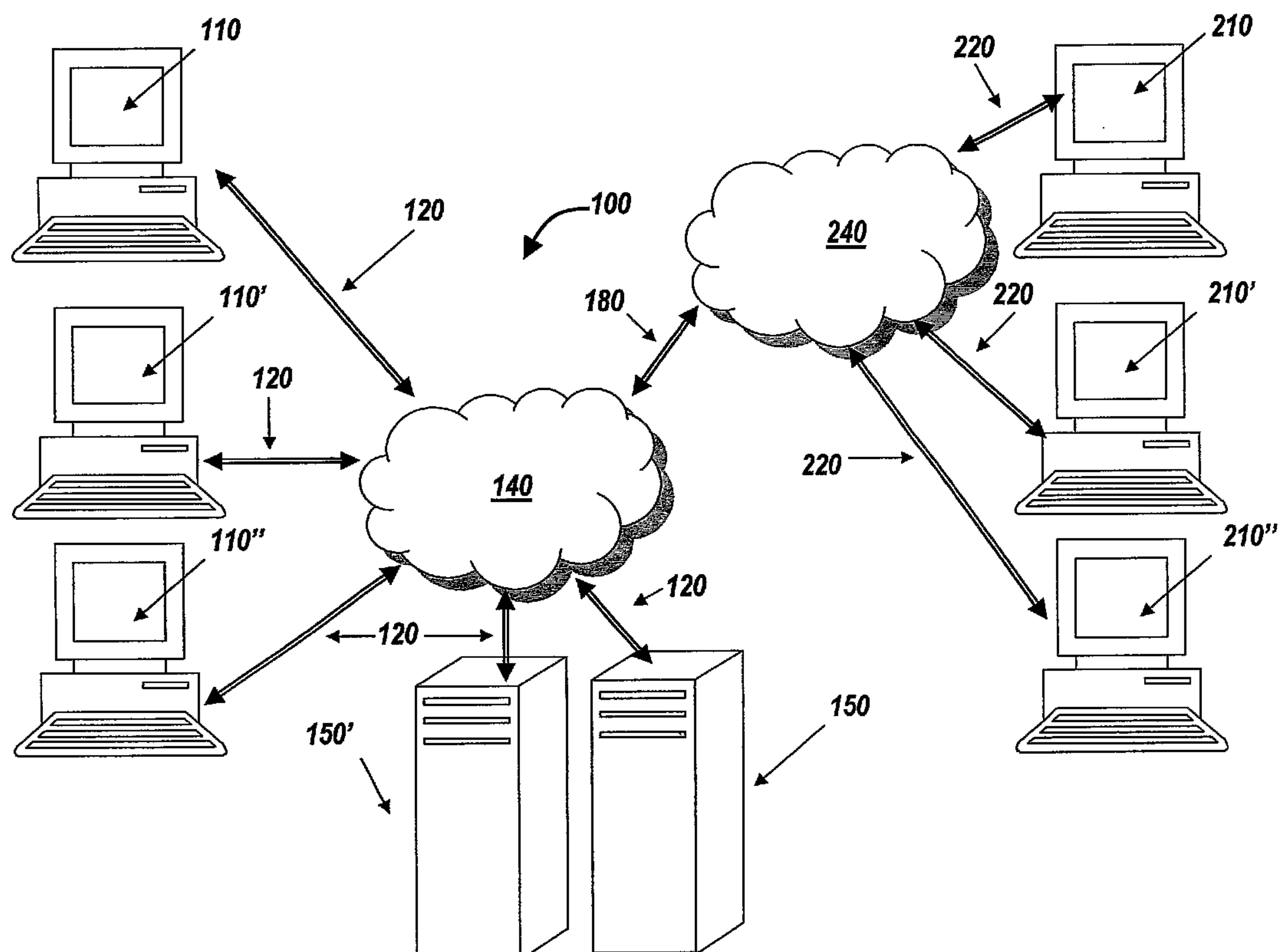




(86) Date de dépôt PCT/PCT Filing Date: 2005/11/17  
(87) Date publication PCT/PCT Publication Date: 2006/05/26  
(85) Entrée phase nationale/National Entry: 2007/05/18  
(86) N° demande PCT/PCT Application No.: US 2005/041773  
(87) N° publication PCT/PCT Publication No.: 2006/055758  
(30) Priorité/Priority: 2004/11/17 (US60/628,806)

(51) Cl.Int./Int.Cl. *G06F 17/30* (2006.01)  
(71) Demandeur/Applicant:  
IRON MOUNTAIN INCORPORATED, US  
(72) Inventeurs/Inventors:  
BLUMENAU, STEVEN, US;  
BARNES, MATTHEW, US;  
PASSMORE, DALE, US  
(74) Agent: GOWLING LAFLEUR HENDERSON LLP

(54) Titre : SYSTEMES ET PROCEDES POUR GERER DES ACTIFS NUMERIQUES  
(54) Title: SYSTEMS AND METHODS FOR MANAGING DIGITAL ASSETS



(57) Abrégé/Abstract:

Systems and methods for managing digital assets in a distributed computing environment are described. Meta-data for the digital assets is stored separately from the digital assets. Meta-data for some of the digital assets is copied and stored at a central location. Meta-data for the digital assets is generated by clients of the system.



## (12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
26 May 2006 (26.05.2006)

PCT

(10) International Publication Number  
**WO 2006/055758 A3**

(51) International Patent Classification:  
**G06F 17/30** (2006.01)

(74) Agent: **MARTINSON, Leigh, John**; Choate, Hall & Stewart, Two International Place, Boston, MA 02110 (US).

(21) International Application Number:  
PCT/US2005/041773

(81) Designated States (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(22) International Filing Date:  
17 November 2005 (17.11.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
60/628,806 17 November 2004 (17.11.2004) US

(71) Applicant (*for all designated States except US*):  
**AVALERE, INC.** [US/US]; 139 Newbury Street, Framingham, MA 01701 (US).

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

(72) Inventors; and

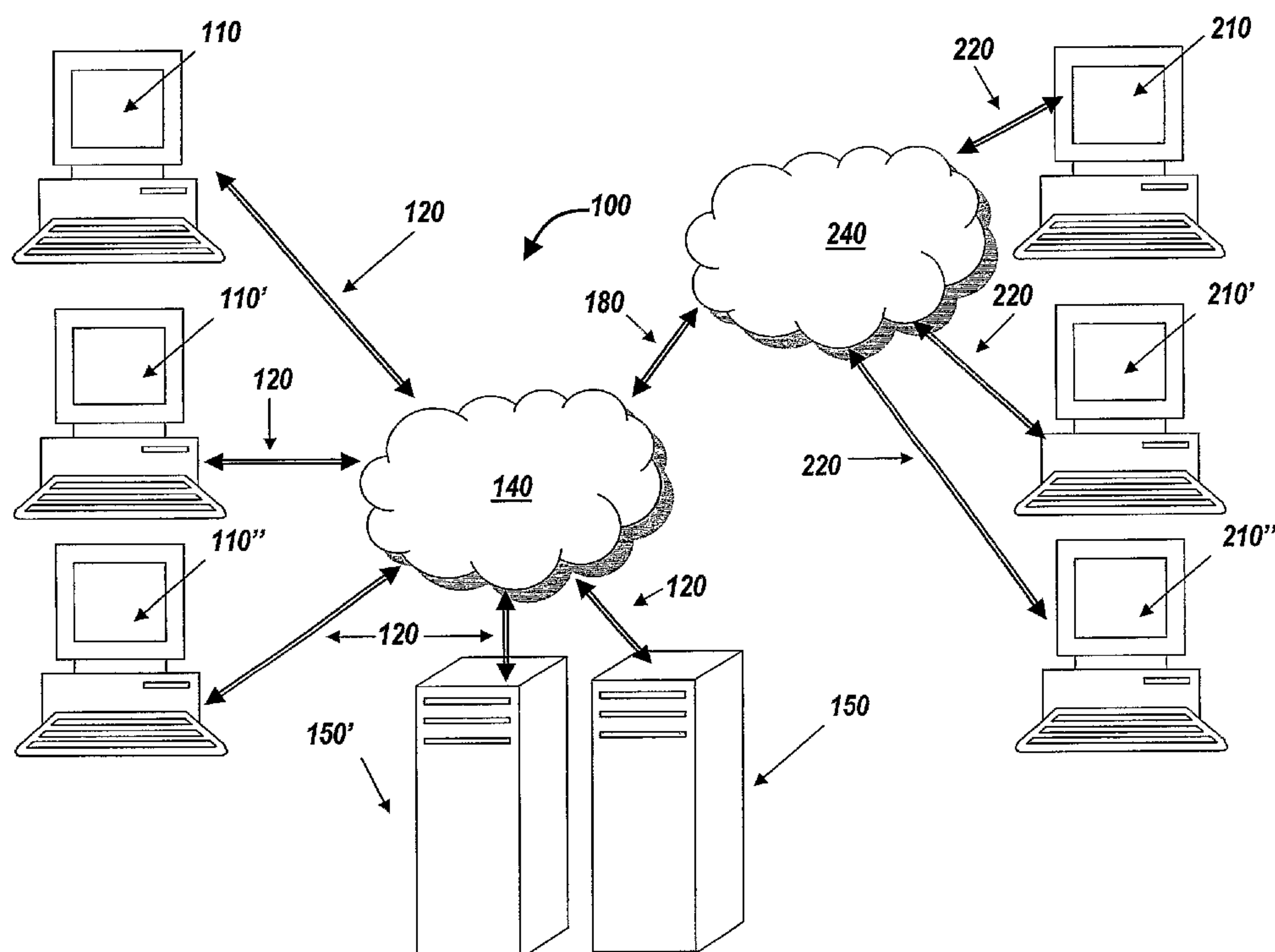
(75) Inventors/Applicants (*for US only*): **BLUMENAU, Steven** [US/US]; 170 Holly Lane, Holliston, MA 01746 (US). **BARNES, Matthew** [US/US]; 74 Josephine Avenue #B, Somerville, MA 02144 (US). **PASSMORE, Dale** [US/US]; 1625 Stoney Point Ct., Colorado Springs, CO 80919 (US).

Published:

— with international search report

[Continued on next page]

(54) Title: SYSTEMS AND METHODS FOR MANAGING DIGITAL ASSETS



(57) Abstract: Systems and methods for managing digital assets in a distributed computing environment are described. Meta-data for the digital assets is stored separately from the digital assets. Meta-data for some of the digital assets is copied and stored at a central location. Meta-data for the digital assets is generated by clients of the system.

WO 2006/055758 A3

**WO 2006/055758 A3**



---

**(88) Date of publication of the international search report:**  
17 August 2006

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*



## SYSTEMS AND METHODS FOR MANAGING DIGITAL ASSETS

### FIELD OF THE INVENTION

**[0001]** The invention relates to managing digital assets in a distributed computing environment.

### BACKGROUND OF THE INVENTION

**[0002]** Centralized document management and other centralized applications can ease digital asset management tasks. However, these tools are expensive, difficult to install and configure, and require end-users to change the manner in the way they work and interact with each other. These solutions are also very dependent upon the end-users to self-enforce corporate governance policies with respect to the digital assets that they create.

**[0003]** The centralized file control mechanism used by these present solutions typically requires end-users to use a burdensome check-in/check-out process to obtain files. However, many end-users prefer not to give up control of their digital assets, are unwilling to sacrifice the ability to use their laptops when they are detached from the corporate network, and resist the workflow requirements of centralized systems. This leaves open the potential for many files located on file servers, laptops, desktops, personal digital assistants (PDAs), and other computing devices to remain outside the controls of the digital asset management system.

**[0004]** Additional drawbacks of current centralized document management tools include: the inability to categorize all digital assets on a storage device; the poor quality of existing categorization techniques when used within a structured context; the inability to provide effective automated control over categorization of digital assets as they are created and changed; the inability to request categorization information from the end-user; the inability to selectively record categorization information based on the conceptual value of the assets; and the lack of category maintenance as assets are copied, moved, renamed, deleted and restored.

[0005] There is, therefore, a conflict between the benefits of centralized file management and end user behavior; a tension which limits the amount of information that will be captured by a centralized document management system.

## SUMMARY OF THE INVENTION

[0006] The following presents a simplified summary of the invention in order to provide a basic understanding of some aspects of the invention. This summary is not an extensive overview of the invention. It is not intended to identify key or critical elements of the invention or to delineate the scope of the invention. Its sole purpose is to present some concepts of the invention in a simplified form as a prelude to the more detailed description presented below.

[0007] As a general introduction, the invention includes a computer software system for gathering and recording categorization data when a digital asset (e.g., file, voicemail, instant message log, email, and the like) or a digital asset container (e.g., folder, directory, disk drive, removable storage medium, and the like) is created. The system executes in a pre-emptive multi-tasking environment. In various embodiments, the software system provides the following features: the ability to categorize existing digital assets in a file system; categorizing digital assets upon creation; a structured and adaptable set of terms (i.e., a taxonomy) for categorization; rule-based categorization of digital assets, minimal change or interruption to the end-user while the software system is in use; the ability to gather meta-data about the digital assets from the end-user; categorization meta-data that is independent of the stored digital asset and structured for simple retrieval of the digital asset; segregation of categorization meta-data (e.g., storing only the meta-data for a digital asset is indicated of value); maintaining the meta-data over time; and propagating the meta-data with the digital asset when the digital asset is transmitted, printed, moved, or copied.

[0008] In one aspect, the present invention allows the end-user to leave a digital asset in the location it is most productive to the end-user rather than moving everything into a centralized repository. In today's distributed and mobile corporate world, it is important that information reside locally with the end-user to enhance



productivity while remaining under control of the corporation. To this end, the present invention provides a means to categorize digital assets at the point of creation with little or no work on the end-user's behalf. The location of digital assets is tracked without requiring the digital asset by stored at a central location.

[0009] Once digital assets are categorized, the present invention allows a policy application to the digital assets. One benefit provided by this feature is that corporations can apply policies to digital assets according to a centralized policy. For example, a corporation decides upon a behavior such as privacy for specific human resource digital assets. That policy is then applied to all digital assets of that type, regardless of the form of the digital assets (e.g., files, email, instant message (IM) logs, etc) that are tagged as human resources digital assets.

[0010] In order to provide the necessary level of control and management, operations performed on the digital assets can be audited. For example, using the categorization the end-user or administrator can set the level of audit to be performed. In one embodiment, a low level of audit would simply keep track of copies and relationships while a high level of audit would keep track of every operation that took place on the digital asset and for the length of time required to perform the operation.

[0011] Other features provided by the present invention include, but are not limited to retention/deletion of digital assets, automatic creation of copies of digital assets, prevention of operations on digital assets, expiration of archived copies of digital assets, storing meta-data separate from the digital asset, prevention of restoration of expired digital assets, searching the digital assets using virtual folders having labels based on the meta-data, copy tracking of digital assets, combining meta-data tags, propagating the meta-data tags with the digital assets, providing an adaptive taxonomy used to create meta-data for the digital assets.

[0012] Retention/Deletion: Each type of digital asset has controls on the minimum length of time that the digital asset is stored and possibly the maximum length the digital asset can be stored. This translates to storing digital assets for a given period of time and then either archiving the digital asset or destroying the digital asset. In certain embodiments, the present invention provides this feature.

[0013] Automatic Copy: There are a number of reasons to make automatic copies of digital assets. For example, the end-user or administrator could set a policy to make a copy of financial digital assets. Another example would be to make a copy of digital assets from a local storage device to a centralized storage device so the digital assets can be archived (i.e., backed-up). In certain embodiments, the present invention provides such functionality.

[0014] Prevention of Operations: In various embodiments, the present invention applies a policy to block certain operations from being performed on a digital asset. For example, specific digital asset can be prevented from being transmitted outside of the company. Another example of a policy is to prevent specific digital assets from being copied to specific devices such as removable media, e.g., USB devices. Policies can also be applied based on the role of the end-user to provide role based access control to certain digital assets.

[0015] Expiration of Archived Copies: In certain embodiments, specific expiration policies are applied to digital assets. When copied to the storage medium for archiving, these policies are copied along with the digital assets. Should an attempt to restore the copies from the storage medium, the policies applied to the copies prevent their restoration. In other embodiments, an encryption key that was used to encrypt the copies prior to storage on the medium is destroyed after an assigned expiration date.

[0016] Additional features and aspects of the invention are described in greater detail below.

#### **BRIEF DESCRIPTION OF THE DRAWINGS**

[0017] The advantages of this invention, described above, and further advantages, may be better understood by referring to the following description in conjunction with the accompanying drawings, in which like numerals indicate like structural elements and features in various figures. The drawings are not necessarily to scale, emphasis instead being placed upon illustrating the principles of the invention.

[0018] FIG. 1 shows an embodiment of a distributed computing environment (DCE).



[0019] FIG. 2 shows an embodiment of a client of the DCE of FIG. 1 constructed according to principles of the invention.

[0020] FIG. 3 shows an embodiment of a server of the DCE of FIG. 1 constructed according to principles of the invention.

[0021] FIG. 4 shows an embodiment of an adaptive taxonomy that incorporates principles of the invention.

[0022] FIG. 5 shows a flow chart of an embodiment of a method of generating meta-data for a digital asset using the client software of FIG. 2 that is operating according to principles of the invention.

[0023] FIG. 6 shows an embodiment of a method of providing meta-data using a graphical user interface according to principles of the invention.

[0024] FIG. 7 shows an embodiment of a method of generating a digital identifier for a digital asset according to principles of the invention.

[0025] FIG. 8 shows an embodiment of a method of tracking copies of a digital asset according to principles of the present invention.

[0026] FIG. 9 shows an embodiment of a method of locating a digital asset in the distributed computing environment according to principles of the invention.

[0027] FIG. 10 shows an embodiment of a graphical display of a locate result constructed according to principles of the invention.

[0028] FIG. 11 shows an embodiment of a method of expiring a digital asset according to principles of the invention.

[0029] FIG. 12 shows an embodiment of a method of preventing the restoration of an expired digital asset according to an embodiment of the invention.

[0030] FIG. 13 shows an embodiment of a method of performing categorization based access to a digital asset.

[0031] FIG. 14 shows an embodiment of a method of propagating the meta-data with a digital asset.

[0032] FIG. 15 shows an embodiment of a method of creating an alias to a tag of the adaptive taxonomy of FIG. 4.



[0033] FIG. 16 shows an embodiment of a method of unionizing different meta-data sets for the same digital asset in accordance with principles of the invention.

[0034] FIG. 17 shows an embodiment of a method of identifying digital assets in the DCE of FIG. 1.

### DETAILED DESCRIPTION

[0035] The present invention provides systems and methods for managing digital assets in a distributed computing environment (DCE). The invention relates generally to the collection, recording and maintenance of meta-data that identifies and categorizes stored digital assets for later location, retrieval and application of business controls. The term meta-data and asset identification tag are used synonymously throughout the specification to refer to the information that is created and used by the present invention to identify and categorize digital assets. Although some of the meta-data created by the present invention corresponds to known meta-data of a file system (e.g., the i-node associated with a file by the Unix operating system or a Master File Table Record used by the WINDOWS operating system, manufactured by Microsoft Corporation of Redmond, Washington) the meta-data of the present invention supplements and extends the known file system meta-data.

[0036] With reference to FIG. 1, a distributed computing environment (also referred to as a client/server system) 100 in which principles of the present invention can be practiced includes one or more clients 110, 110', 110" (hereinafter each client or plurality of clients is generally referred to as 110) in communication with one or more servers 150, 150' (hereinafter each server or plurality of servers is generally referred to as 150) via communications network 140 through communications links 120. The communications network 140 can be a local-area network (LAN), a medium-area network (MAN), or a wide area network (WAN) such as the Internet or the World Wide Web. The communication links 120 can be a variety of connections including standard telephone lines, LAN or WAN links (e.g., T1, T3, 56kb, X.25), broadband connections (e.g., ISDN, Frame Relay, ATM), and wireless connections (e.g., IEEE 802.11). The clients 110 and servers 150 communicate through the network 140 using a variety of communication protocols (e.g., TCP/IP, IPX, SPX, NetBIOS, NetBEUI, and direct asynchronous protocols).

[0037] Additionally, the clients 110 can communicate with other clients 210, 210', 210'' (hereinafter each other client or plurality of other clients is generally referred to as 210), which can be connected to a second network 240, through a communication link 180 that connects network 140 to the second network 240. The protocols used to communicate through communications link 180 can include any variety of protocols used for long haul or short transmission. For example, TCP/IP, IPX, SPX, NetBIOS, NetBEUI, SONET and SDH protocols.

[0038] The client 110 can be any personal computer, Windows-based terminal, Network Computer, wireless device, information appliance, RISC Power PC, X-device, workstation, minicomputer, main frame computer, cellular telephone or other computing device that provides sufficient faculties to execute client software and an operating system. Client software of the invention facilitates the creation of meta-data that identifies, categorizes, and characterizes the digital assets generated and stored by the client. As used herein, digital asset refers to any digital file that can be stored in a storage medium. Examples of digital assets can include, but are not limited to, files, emails, instant messages (IM), audio files, video files, profiles, drivers, programs, and other electronic embodiments of information.

[0039] The server 150 can be any type of computing device that is capable of communication with the client 110. For example, the server 150 can be a traditional server computing device, a web server, an application server, a DNS server, or other type of server. Additionally, the server 150 can also be a client 110 (e.g., in an ad-hoc or peer-to-peer (P2P) network arrangement). One purpose of the server 150 is receiving, storing, and managing meta-data associated with the digital assets of the clients 110. The sever 150 can also provide a means to modify and update a taxonomy used to categorize and create meta-data for the digital assets, request that the client perform operations on its stored digital assets, and generate reports on the state of the stored meta-data. One example of a server 150 that can be used with the invention is a DELL server classes computer having 1 gigabyte of RAM, dual central processing units, a 250 gigabyte hard drive, and an network interface card. It should be understood that more than one server 150 can be used with the present invention. In such a configuration, functionality can be distributed across the servers 150 or each server 150 can provide a full suite of functionality.



[0040] FIG. 2 depicts a conceptual block diagram of a client 110 of the distributed computing environment 100. Each client 110 typically includes a processor 200, volatile memory 204, an operating system 208, client software 212, a persistent storage memory 216 (e.g., hard drive or external hard drive), a network interface 220 (e.g., a network interface card), a keyboard 224 or virtualized keyboard in the case of a PDA, at least one input device 228 (e.g., a mouse, trackball, space ball, light pen and tablet, touch screen, stylus, and any other input device), and a display 232. The operating system 116 can include, without limitation, WINDOWS 3.x, WINDOWS 95, WINDOWS 98, WINDOWS NT 3.51, WINDOWS NT 4.0, WINDOWS 2000, WINDOWS XP, WINDOWS VISTA, WINDOWS CE, MAC/OS, Java, PALM OS, SYMBIAN OS, LINSPIRE, LINUX, SMARTPHONE OS, and the various forms of UNIX.

[0041] The client software 212 is in communication with various components of the client 110 to provide features of the invention. In one embodiment, the client software 212 includes an agent 250, one or more filter drivers 254, and one or more plug-in modules 258. It should be understood that the client software 212 can include some or all of the components shown and described. As a general overview, the client software 212 provides a means to create, edit, maintain, update, revise, modify, and produce meta-data that provides categorization and identification of digital assets. The meta-data is associated with some or all of the digital assets created or stored on the client 110 and is used to provide tracking, locating, searching, and other features and aspects of the invention.

[0042] The agent 250 operates in the “user space” of the operating system 208 as do a various plug-in (also referred to as Add-in) modules 258. The agent 250 and plug-ins 258 are in communication with the various filter drivers 254, which operate in the “system space” of the operating system. Although shown in user space, it should be understood that in certain embodiments, the agent 250 can operate in the system space as well. The cooperation of the agent 250, the filter drivers 254, and the plug-in modules 258 provide the end-user of the client 110 with the features and operational characteristic of the invention. These features can be invisible to the end-user (e.g., automatic categorization of digital assets) or require end-user input through a graphical user interface (GUI) (e.g., end-user categorization). For example, when a

request to create a folder is executed, the filter driver 254 intercepts the command. The filter driver 254 communicates with the agent 250. In response, the agent 250 displays a graphical dialog and asks the end-user for meta-data information (e.g., categorization information). In one embodiment, the client software 212 also interacts with a file system filter driver 258 that is provided as part of the operating system 208. In another embodiment, the client software 212 replaces the file system filter driver 258 provided by the operating system 208.

[0043] During certain modes of operation, the client software 212 intercepts file system commands and performs various functions of the invention in response thereto. For example, prior to adding a new digital asset to the file system of the client 110 the client software 212 intercepts the file system command to create the digital asset and requires the end-user to provide at least a portion of the meta-data (e.g., categorization information) associated with the digital asset. After the client software 212 applies the meta-data, the digital asset is added to the file system of the client 110. Another feature the client software provides is the generation of a digital identifier that is associated with digital asset as part of the meta-data. The categorization information and digital identifier form, in one embodiment, the meta-data that is associated with the digital asset. Another exemplary feature provided by the client software 212 is to perform a search or locate. The end-user of the client issues a search or locate command, the client software 212 intercepts this command and provides a “virtualized” view of the contents of the file system of the client 110. Each of these examples is explained below in more detail.

[0044] The associated meta-data for each digital asset may or may not be forwarded to the server 150 via network interface module 220 and communications link 120. Whether the meta-data for the digital asset is transmitted to the server for storage depends on the categorization and rules applied to the digital assets. This provides for granular control of certain digital assets of interest.

[0045] With reference to FIG. 3, an embodiment of a server 150 for user in the distributed computing environment 100 is described. The server 150 includes a processor 300, a volatile memory 304, an operating system 308, server software 312, persistent storage memory 316, a network interface 320, a keyboard 324, at least one input device 328 (e.g., a mouse, trackball, space ball, bar code reader, scanner, light



pen and tablet, stylus, and any other input device), and a display 332. The server operating system can include, but is not limited to, WINDOWS XP, WINDOWS 2000 SERVER, WINDOWS 2000 ADVANCED SERVER, WINDOWS NT SERVER, WINDOWS NT SERVER ENTERPRISE EDITION, MACINTOSH OS X SERVER, LINUX, UNIX, SOLARIS, VMWARE, and the like.

[0046] A central repository 336 (e.g., a database) is in communication with the server 150. Although shown as separate from the server 150, it should be understood that the central repository 336 can be integral with the server 150 or located elsewhere within the distributed computing environment 100. The central repository 336 is configured to store meta-data associated with certain digital assets. In one embodiment, the digital assets and their associated meta-data are stored at the clients 110 and a copy of the associated meta-data is stored at the central repository 336. This provides a “decentralized” digital asset management system, which enables certain features and advantages of the invention. For example, by not storing the digital assets themselves at the central repository 336 the end-users are not required to check-out and check-in the digital assets in order to perform operations on the digital assets.

[0047] Additionally, the communication link 120 that connects the client 110 to the server 150 does not need to be maintained thereby tethering the client 110 to the server 150. Said another way, the communication link can be established on an “as-needed” basis. This feature allows the end-user to work “off-line” with the digital assets of interest and upload changes to the meta-data when a connection to the central repository 336 is established. Additionally, changes to the meta-data for a digital asset can be downloaded from central repository 336 when a connection is established. Also, various policies associated with the meta-data of the digital assets can require performance of specific tasks when the client 110 connects to the server 150. it should be understood that when the client 110 connects to the server 150 these tasks are executed.

[0048] In certain embodiments, the server software 312 provides a means to perform certain features of the invention. For example, the server software 312 allows an administrator to create and modify an adaptive taxonomy that is used to create categorization information for a digital asset. Also, the server software 312

propagates different meta-data sets for the same digital each to each client 110 having a copy of the digital asset. The clients 110, in turn, perform a union of the different meta-data sets. In other embodiments, the server software 312 cooperates with the client software 212 to enable other features of the invention. For example, an administrator can issue a command using the server software 312 to copy certain digital assets to a central location in an effort to produce documents required in litigation. An example of a function that is performed by the client software 212, but can also be provided by the server software 312 is the ability to perform a union of meta-data for a digital asset and propagate a selected characterization for that digital asset. Each of these features is described in more detail below

[0049] With reference to FIG. 4, an exemplary adaptive taxonomy 400 of the invention is described. As used herein, taxonomy refers to a hierarchical structure of tags used to provide a method of organizing digital assets. Conceptually, a taxonomy can be thought of as a tree structure having a root node 410, a plurality branches 420 connecting leaf nodes 430. Each leaf node 430 can have further braches 420 that connect the leaf nodes 430 to sub-leaf nodes 440 and so on. As used with reference to the taxonomy 400, the terms node and tag are synonymously.

[0050] Each node 430 and sub-node 440 can be applied to a digital asset as a tag that is part of the meta-data for the digital asset. The tag that is used to identify and categorize the digital asset. When used properly, a taxonomy 400 not only helps an organization organize digital assets but the taxonomy also helps identify types of digital assets. Policies can also be associated with each node 430 and sub-node 440 of the taxonomy 400. Applying a node 430 or sub-node 440 as a tag of the taxonomy to a digital asset also associates the policy for that node to the digital asset. Examples of policies can include, but are not limited to, restricting access to a digital asset based on the role and/or identity of the end-user of the client 110, restricting replication actions based on the destination of the copy of the digital asset or the present location of the digital asset, and when the digital asset is removed from the client 110.

[0051] Although a taxonomy 400 is a powerful organizational tool, a rigid taxonomy restricts the flexibility of digital asset characterization. To that end, the invention provides a mechanism in which modifications to the taxonomy 400 can be made be the end-users of the clients 110 on an individual level without requiring



modifications to the general taxonomy 400. Also, if a change to the general taxonomy 400 is required, the invention provides a mechanism for propagating the changes to the taxonomy 400 to the clients 110.

[0052] To accomplish these features, the invention provides the functionality to create an “alias” for a node 430 or sub-node 440 in the taxonomy. As used herein, an alias refers to an alternate name for the same tag in the taxonomy 400. For example, the term “CV” (Curriculum Vitae) is used in many parts of the world to have the same meaning as “resume” is used in the United States. In the taxonomy 400, a tag 440 is labeled “Resume” and has an alias 450 labeled “CV” associated with it. Essentially, the alias 450 points to the associated tag 440 and has the categorization and policy information as the tag 440. As will be described in more detail below, the alias 450 can be a local alias meaning that is available only to a specific client 110 or the alias can be a global alias meaning that the alias is available to all clients.

[0053] The invention includes functionality implemented, in one embodiment, by the server software 312 to promote an alias 450 to a tag 440. The promotion does not change how the alias 450 has been used previously. That is, digital assets that were tagged with the alias 450 are still governed by the same categorization and policy information of the alias 450. It should be understood that the transition from an alias to a tag 440 allows for the modification of the policies associated with the alias 450. Further details of the adaptive taxonomy 400 are provided below.

[0054] With reference to FIG. 5, the method 500 for generating meta-data for a digital asset is shown and described. In one embodiment, a client 110 executing client software 212 generates a digital asset. The client software 212 intercepts a create or a save command for the digital asset and generates (step 510) an asset identification tag. The asset identification tag is the meta-data that is associated with the digital asset. Further, the client software 212 associates (step 520) the asset identification tag with the digital asset. The client stores (step 530) asset identification tag. Optionally, the asset identification tag is transmitted (step 540) to server 150 for storage in the central repository 336.

[0055] In one embodiment, generating an asset identification tag (step 510) is performed when the digital asset is stored at the client. In another embodiment, the asset identification tag is created when the end-user begins to create a new digital

asset. For example, if the end-user of a client creates a new folder or directory for storing digital assets, the client software 212 examines any rules that related to the creation of the folder to categorize the new folder based first on the device on which the folder is being created, next based on the application creating the folder, and lastly the end-user creating the folder. However, if required, the end-user can be prompted to provide categorization information via an end-user interface. The categorization data is saved and the folder is created within the file system of the device. It should be understood that once a categorization data is applied to a digital asset, the categorization may be changed at a later time, if the associated rules allow. This allows for recategorization of certain digital assets while preventing recategorization of other digital assets. The terms rule and policy are used interchangeably throughout the specification.

[0056] In various embodiments, application rules define the set of categorized digital assets (e.g., taxonomy tags) that can be stored with a directory or file when that directory or file is created by an application. Application rules consider the name and context of the digital asset (binary name, binary versions, process name, window titles, and the link) and the name of the directory being created. From this data a set of taxonomy tags are determined and returned as the list of is tags for this digital asset.

[0057] Device rules define the set of taxonomy tags that can be applied to a directory or file when that directory or file is created by or stored on a particular device. Rules can be defined for device classes (e.g., local fixed device, network device, removable devices), individual storage devices or input devices. Similarly, end-user rules define the set of taxonomy tags that can be associated with a directory or file when that directory or file is created or changed by the end-user. User rules can consider the end-user's name, the end-user's role, the end-user's location or any other data that can be retrieved from a local or directory based end-user configuration.

[0058] By applying rules and categorizations to folders, directories, end-users, and devices, automatic and inheritance based categorization of digital assets is achieved. For example, if a word document is stored in a specific directory, the client software 212 applies the taxonomy tag indicated by the rules and categorization of the directory that stores the word document. Further, if a one or more uncategorized



digital assets are moved into a categorized directory those digital asset inherent the categorization of the directory. Such a feature allows for the categorization of digital assets existing on the client 110 prior to the installation of the client software 212.

**[0059]** The following example is designed to illustrate one embodiment of categorizing a digital asset. The example should not be read to limit the scope the invention. Assume that an end-user John Smith who works the finance department creates an Excel file in the “\\finance\john\budget” folder of his home directory that was categorized using the taxonomy tags 430 and 440. The client software 212 creates meta-data that contains various categorization information based on John’s identify such as: data created, author, department, etc. The client software 212 can also add meta-data resulting from the rules associated with the “budget” folder (or its parent folder Finance) such as confidential, marked for compliance, do not delete, do not email, and the like. The level of meta-data granularity can be further augmented with input from John using the graphical user interface if desired by John or required by the rules.

**[0060]** A method 600 of providing meta-data information using the graphical user interface is shown and described with reference to FIG. 6. The client software 212 provides (step 610) the end-user of the client 110 with a graphical display having one or more dialog boxes, lists, or radio buttons. The end-user manipulates the graphical user interface to provide meta-data that is associated (step 620) with the digital asset. Manipulation can include, but is not limited to, selecting a taxonomy tag 430 to apply to the digital and the like.

**[0061]** In addition to generating categorization information as part of the meta-data for a digital asset, the client software 212 can generate a digital identifier for each digital asset. One embodiment of a method 700 for generating such a digital identifier is shown and described with reference to FIG. 7. For example, during a save operation the client software 212 analyzes (step 710) the contents of the digital asset using a hash function. In one embodiment, the client software 212 analyzes the text of the digital asset. In other embodiment, additional or other elements of the digital asset are analyzed. For example, the categorization information can also be included in the analysis, or in the case of an email or instant message the sender and recipient of the email or instant message. Examples of hash function that can be used by the

client software 212 include but are not limited to MD5 (IETF RFC1321) and SHA1 (IETF RFC3174).

**[0062]** Also, the meta-data can include a list of keywords that are a part of the digital asset. One method of generating the list of keywords for the digital asset is to analyze the digital asset and record words of importance. It should be understood that certain words will not be recognized as keywords. For examples, articles such “a”, “an”, and “the”, or pronouns, will not be recorded as keywords. Various known techniques can be used to generate the list of keywords for the digital asset.

**[0063]** The combination of the digital identifier, keywords, and the categorization information described above, or respective combinations of portions of each create the asset identification tag (i.e., meta-data) for the digital asset. As previously stated, the asset identification tag is associated (step 520) with the digital asset. Association can include creating a “hidden” file that stores the meta-data that is permanently linked to the digital asset. As used herein, permanently linked refers to an association that can not be removed regardless of the transmission, moving, or copying of the digital asset. For example, if a digital asset is emailed to another end-user the associated asset identification tag is emailed as well. The propagation of an asset identification tag will be described in more detail below.

**[0064]** In one embodiment, storing (step 530) the asset identification tag includes storing the asset identification tag in the persistent storage 216 of the client 216. The asset identification tag can be stored in the same shared storage area as the digital asset. Alternatively, the asset identification tag is stored separate from the digital asset. For example, in a different dedicated memory location or another storage device.

**[0065]** In order to determine whether to transmit (step 530) a copy of the asset identification to the server 150, the meta-data of the digital asset is resolved to one of three levels: (1) unmanaged; (2) managed; or (3) records managed. In one embodiment, if the digital asset is resolved to be unmanaged then the asset identification tag is not stored by the client 110 or the server 150. However, if the asset identification tag is resolved to be managed then the asset identification tag is stored locally at the client 110. Finally, if the asset identification tag is resolved as records managed a copy of the asset identification tag is transmitted to the server 150



to notify the server software 312 of the existence of the digital asset. It should be noted that actually the digital asset is not transmitted to the server 150, but instead the digital asset is stored locally at the client 110. Although described as having three levels of resolution, it should be understood that a fewer number or greater number of levels are possible.

**[0066]** The advantages of having a class of “managed” digital assets and a class of “records managed” digital asset is to treat the digital assets in the manner similar to the other assets of a corporations. For example, look at the difference between pencils and computers in a corporation. As with any asset in a corporation, pencils need to be managed. In the case of pencils, the corporation likes to know how many have been ordered and have a general idea of when to order additional pencils. The corporation is typically not concerned with who has a pencil or how many pencils are owned by each person. In contrast to pencils, the corporation wants to know exactly which end-user has each computer and where the computer is located. This analogy translates directly to digital asset. For example, “managed” digital asset can be mapped to pencils and “records managed” digital assets can be mapped to computers. An example of a managed digital asset can be a voice mail from potential new client. An example of a records managed digital asset can be an invention disclosure. By using a leveled approach to digital asset classification, the centralized repository needs only to track a percentage of the digital assets in the distributed computing environment 100 instead of all the digital assets. The digital assets that are not tracked by the centralized repository are tracked by the clients 110. As previously explained, the clients 110 track each of the assets that are stored locally a the client.

**[0067]** One way to determine which digital assets are unmanaged, managed, or records managed is to use the taxonomy tags 430. As previously stated, each digital asset is associated with at least one tag 430 of the taxonomy 400. The taxonomy tag includes policy information (e.g., rules) and a digital asset classification level. Other meta-data entries can also be used to determine whether a digital asset is unmanaged, managed, or records managed. For example, the creation date of the digital asset can be used.

**[0068]** In one embodiment, any digital asset that exists on a client 110 when the client software 212 is installed is automatically categorized as unmanaged. As a

result, no meta-data entry on the client is created for these digital assets. The taxonomy 400 can include a tag 430 in it that is labeled “unmanaged” that includes associated policies that are applied to unmanaged digital assets. Typically a small set of policies is used. For example, an expiration date (i.e., expire the asset in 1 year) and also a location control policy that does not allow the digital to be moved, copied, emailed, or otherwise transferred from the current client 110. Similarly, every other tag 430 of the taxonomy 400 can include a rule that creates a meta-data entry giving the digital asset a level of either managed or records managed. For example, if a digital asset is associated with the IP tag 430 the meta-data for the digital asset indicates that the digital asset is classified as managed. Further, if the digital asset is associated with the disclosure sub-tag 440 the meta-data for the digital asset indicates that the digital asset is records managed and a copy of the meta-data is transferred to the server 150 for storage.

[0069] In addition to a digital identifier and categorization information, the meta-data for a digital asset can include a list of operations performed on the digital asset by the client 110. This information can be thought of as an audit history and is useful for many things. For example, determining the number of copies of a digital asset that exists, which end-user created the copies, what application created the copies, what is the source of the copy of the digital asset, and which devices stored the copies. The resulting copy not only includes all the contents of the original digital asset, but also the meta-data for the original digital asset, which include the digital identifier of the original file. Said another way, when a copy operation is performed both the contents of the digital asset and its associated meta-data are copied.

[0070] It should be understood that the audit information for the same digital asset existing on the different clients 110 can have different contents. If the digital asset is records managed, the audit information for each asset is transferred to the server 150 as part of the meta-data for the asset. The server software 312 performs a union of the audit information, propagates the unionized audit information to the clients 110, and instructs the client software 212 to remove the local copy of the pre-unionized meta-data and replace the pre-unionized meta-data with the unionized meta-data.

[0071] Various methods for copying various digital assets are known. Methods for files and email, both of which digital assets, are described below. The most



straight forward way to create a copy of a digital asset is to print the digital asset onto paper. This creates a “hard copy” of the digital asset. It is important to track the printing of digital assets for a number of reasons. One reasons is for expiration purposes. It is desirable to know that if a digital asset was printed the day of its expiration that the paper copy was also destroyed. Another reason for tracking copies of digital assets, is to monitor which end-users are access and copying which digital assets.

[0072] Also, there are many different ways that an electronic copy of a file can be created by the end-user of the client 110. For example, the end-user can execute the “copy” command in windows explorer and then execute a “paste” command in another location. This causes a copy of the content of the file to be created. Even though there are many methods to create a copy of a file, the actual create of the new copy must go through the file system of the client 110. As a result, filter driver 254 is used to identify when a new file for the file system is created.

[0073] It is also important to determine when an open file of the file system is written to. For example, an application might open a first file “A” for reading and a second file “B” for writing. The application under the instruction of the end-user copies the contents of the first file A to the second file B. In this example, the second file B was not created it was only updated with the contents of the first file A. In another embodiment, file B is created as a new file and the contents of file A are copied into file B.

[0074] Similar to files, there exists a number of methods that can be employed to create a copy of an email. The simplest method is to “copy” an email and then “paste” it using the functionality provided by the email application of the client 110.

[0075] Another way to create a copy of an email is to copy the folder or the email application file that stores the email or emails for the email application. Within an email application, an email can be stored within a “folder” of the email application to provide a means to organize the end-user’s email. The folders and emails that are displayed to the end-user of the email application are stored in files or directories of the client 110, which may be file system folder . Using MICROSOFT OFFICE OUTLOOK as an example, the application creates and uses the .OST and .PST files for holding the definitions of the folders and the emails shown to the end-user of the

application. One method the end-user can use to create a copy of email is to export the email out of the email application. Outlook provides an interface that allows one to export information. Using this feature, one can put the email into a text file, excel spreadsheet or even a .PST file thereby creating a copy of the email.

[0076] Another method of creating a copy of email is to simply copy the .OST and/or the .PST file outside of the email application. In this case, the email application is not necessarily executing on the client 110 during the copy operations. This operation is similar to the copying of a file from the file system as described above.

[0077] With respect to FIG. 8, a method 800 of tracking a copy of a digital asset in the distributed computing environment is shown and described. In one embodiment, the method includes determining (step 810) if a copy of the digital asset is created, generating (step 820) a meta-data entry for the original digital asset that indicates a copy was made, and updating (step 830) the stored meta-data for the digital asset.

[0078] The determining (step 810) can be accomplished in many ways. In one embodiment, the meta-data of the digital asset being created is compared to a list of known meta-data stored on the client 110 or server 150. In another embodiment, only a portion of the meta-data is used to do the comparison (e.g., the digital identifier).

[0079] In another embodiment, the filter driver 254 or the plug-in 258 monitors the action of the applications executing on the client with respect to file I/O. By monitoring an application and its threads, the client software 212 can determine what files are being opened for reading and what files are being open for writing. For example, if an application has opened file A and file B for input and file C for output. File C inherits, as previously described, all the meta-data (e.g., control policies and the like) from both file A and file B and associates them with file C. This method addresses the case of either creating file C as a new file or opening an existing file C for write. Once the first I/O is completed to the output, the meta-data will be updated to the union of file A and file B as described in more detail below. Further, if file C is stored in a folder having an applied taxonomy tag 430, resulting meta-data is the union of file A, file B, and the folder.



[0080] The generation (step 820) of meta-data can be accomplished in various ways. For example, when a print(from the perspective of the client software 212 is essentially creating a paper copy of the digital asset) is executed meta-data about the print is add to the meta-data of the printed digital asset. This meta-data can include various combinations of the date and time the digital asset was printed, which end-user printed the digital asset, which digital assets were the source of the printed digital asset, the digital identifier from the source digital assets, and what printer generated the paper copy of the digital asset.

[0081] Also, various methods of updating (step 830) the meta-data for the digital asset can be used. For example, in the case of a managed digital asset the meta-data previously stored about the digital asset is refreshed with the copy meta-data. In the case of a records managed digital asset, after the locally stored meta-data is updated the updated meta-data is transmitted to the server 150 for storage.

[0082] In certain instances, it is desirable to suspend the creation of meta-data for a digital asset. For example, during the installation of other software applications. Typically, when installing software, a program is executing commands that will cause folders to be created. The end-user could be bombarded with requests for categorization of folders. Because of this, there is a special command that can be executed by the end-user of the client 110 that informs the client software 212 to suspend its operation. After the installation of the software, the client software 212 resumes it's normal operation. Although the operation of the client software 212 can be suspended, the invention monitors what operations are performed while the client software 212 is suspend and records this information to a general audit log for the client 110.

[0083] After creating meta-data for each of the digital assets, the meta-data can be used to provide various features of the invention. Some of these features are provided by the client software 212, some are provided by the server software 312, and some are provided by the cooperation of the client software 212 and server software 312.

[0084] One feature provided by the client software 212 is the ability to locate a digital asset using the keywords and meta-data associated with the digital asset. As used herein, the term locate is used synonymously with the term search. Because each client 110 stores their digital assets locally, the possible solution set to a locate

request is a closed set of digital assets. In essence, when a locate command is executed the full set of possible keywords and meta-data tags that could be used in locate are shown to the end-user of the client 110 as a set of virtualized folders. This removes the requirement from the end-user to input a search term in a search engine if the end-user can not think of a search term. Because most end-users work in a focused area, the number of taxonomy tags and the number of unique keywords stored in the meta-data of the digital assets words are typically limited to the end-users focused work area.

[0085] With reference to FIG. 9, a method of locating a digital asset in the distributed computing environment 100. In one embodiment, the method includes receiving (step 910) a search command from the end-user of the client, identifying (step 920) the taxonomy tags 430 associated with the digital assets that are stored locally at the client, and displaying (step 930) one or more folders to the end-user of the client 110. The folders include labels that are the identified taxonomy tags 430.

[0086] Receiving (step 910) a search or locate command from the end-user of a client 110 can be accomplished in various ways. For example, the end-user can select a hot key (e.g., F12) on a keyword. In another embodiment, the end-user can select a portion of a digital asset and right-click on the selected portion. As a result, a menu is displayed to the end-user that includes a locate menu item. Additionally, the end-user can select a search command from a start menu option.

[0087] Various means of identifying (step 920) the taxonomy tags 430 associated with the digital assets of the client 110 are contemplated. In one embodiment, a scan is performed of all the digital assets stored at the client 110 to determine which taxonomy tags 430 are associated with the digital assets. In another embodiment, the end-user can supply a search term to the locate function. As a result, the identified digital assets include the provided term in their associated meta-data. Alternatively, the provided term is used to exclude taxonomy tags 430. In addition to identifying taxonomy tags 430, the client software 212 can identify the keywords in the associated meta-data for the digital assets. Also, a combination of taxonomy tags 430 and keywords can be used.

[0088] Once the taxonomy tags 430 and/or keywords are identified, the client software 212 generates a virtualized file system view of the associated digital assets



and displays (step 930) to the end-user. In one embodiment, the familiar graphical “explorer” interface is shown to the end-user. With reference to FIG. 10, the explorer view 1000 depicts one or more folders 1004 and/or files to the end-user. The virtual folders 1004 include a label that is one of the identified taxonomy tags 403 or keywords. The virtual folders 1004 are not the actual file system folders. Creation of the virtual folders is accomplished by the client software 212.

[0089] Selecting one of the virtual folders 1004 results in another display of another set of virtual folders. In essence, the system provides a means to “drill down” into meta-data of the digital assets to locate a desired digital asset. By selecting a displayed virtual folder, the client software 212 is in essence performing another search using only the set of digital assets selected from the first search.

[0090] Another feature enabled by the meta-data and client software 212 of the present invention is the ability to control and maintain a document expiration policy. By using the associated expiration date that is present in the meta-data for the digital asset, different sets of digital assets can be exposed to an archive system and recorded to separate mediums. For example, all digital assets and only the digital assets having an expiration date in the range of a given week are exposed to the archive system. At the end of that indicated week, the archive tape can be destroyed, thereby destroying the backed-up copies of the digital assets.

[0091] One embodiment of a method 1100 of expiring stored digital assets is shown and described with reference to FIG. 11. The method includes providing (step 1110) a date range using the client software 212, enumerating (step 1120) the digital assets that have an expiration date within the provide range as file system elements, and storing (step 1130) the enumerate assets on a storage medium.

[0092] The end-user provides (step 1100) a date range to the client software 212 using a graphical user interface or a command line entry. In another embodiment, the client 110 includes one or more archive scripts that are executed automatically. The scripts include date ranges used to expose specific digital assets to the archive system. Although described as storing the digital assets having the associated date range, it should be understood that the provided data range can indicate digital assets that are not be exposed. An indicator or flag (e.g., an exclamation point) can be used to indicate the described “not” function. Also combinations of both types of date ranges

can be used to generate the desired set of digital assets. In addition, the meta-data associated with the digital assets can be used to define the set of digital assets that are exposed for archiving.

[0093] Using the provided data range, the digital assets are separated using a virtual file system. In one embodiment, the filter driver 254 creates a virtual file system enumerating (step 1120) those digital assets having an expiration date within the provided date range. Conceptually, the virtual file system acts as a mask over the actual file system of the client. The mask exposes only those digital assets fulfilling the indicated criteria to the archive system.

[0094] The exposed digital assets are copied (step 1130) to a storage medium. The storage medium can be a tape, disk, or other suitable storage medium. In one embodiment, the digital assets that are copied to the storage medium are encrypted prior to be copied to the storage medium. In another embodiment, when an expiration date is assigned to the meta-data of the digital asset the digital asset is encrypted (step 1140) when the digital asset is stored in the file system of the client 110. Digital assets having similar expiration dates can each be encrypted with the same encryption key, which can also be stored on the storage medium or separate from the storage medium. The encryption key is assigned an expiration date. After the expiration of that date, the encryption key is destroyed (step 1150). In another embodiment, a separate encryption key is used to encrypt each digital asset.

[0095] There are many methods that can be used encrypted the digital assets. For example, an application can encrypt the digital assets. Alternatively, each client can have an encrypted file system such as the Microsoft Encrypted File System. In another embodiment, the filter driver 254 or plug-in 258 can perform the encryption

[0096] Another feature enabled by the meta-data and client software 212 of the present invention and in some embodiments the server software 312 is the prevention of the restoration of a previously expired digital asset. In one embodiment, once a digital asset has been expired the actual digital asset is removed from the client 110. However, the meta-data remains at the client 110 and in the case of a record managed digital asset at the server 150. The meta-data can include an entry that the digital asset has been previously expired.



[0097] With reference to FIG. 12, a method 1200 for prevention of restoration of a digital asset is shown and described. In one embodiment, the method includes receiving (step 1210) meta-data associated with a digital asset that was previously created by a client 110, comparing (step 1220) the received meta-data with the stored meta-data on the client 110 and/or the server 150, and preventing (step 1230) the restoration of the digital asset when the received meta-data matches stored meta-data for the digital asset that indicates the digital asset was previously expired.

[0098] As previously explained, the client 110 and the server 150 need not be in constant communication because the digital assets are not stored at the server 150, and further a full copy of the meta-data is stored at the client 110. When a digital asset is restored to a client 110 that did not create the digital asset and thus does not have a meta-data entry to compare the restored asset to, the client 110 establishes a connection to the server 150. Once the connection is established, the client 110 transmits the meta-data to the server 150 where it is compared (step 1220) against the meta-stored stored at the server 150.

[0099] In one embodiment, the comparing step (1220) includes comparing the entire meta-data contents with the list of known meta-data. In another embodiment, a portion of the meta-data is compared to the list of known meta-data. The portion of the meta-data can include, but is not limited to, the digital identifier or a taxonomy tag. During the comparison process, the client 110 can disconnect from the server 150 or maintain the communication link 120 with the server 150.

[0100] When the server 150 finds a match between the restored digital asset meta-data and previously expired digital asset meta-data, the server 150 issues a command to prevent (step 1230) the restoration of the digital asset to the client 110. In one embodiment, the command includes instructions to remove the restore digital asset. In another embodiment, the command includes instruction to not allow the digital asset to be copied to the file system of the client 110.

[0101] Another function provided by the client software 212 of the present invention is the ability to control access to digital assets using the associated meta-data of the digital assets. Using the meta-data that is associated with each digital asset, role based, user based, and, a combination of role based and user based access is provided.

[0102] One embodiment of a method of providing meta-data based access to a digital asset is shown and described with reference to FIG. 13. The method includes receiving (step 1310) a request to access the digital asset, determining (step 1320) a categorization of the digital asset, evaluating (step 1330) any rules associated with the categorization, and allowing (step 1340) access to the digital asset when the determining and evaluating indicate access is allowed.

[0103] In one embodiment, the receiving (step 1310) includes intercepting, by the filter driver 254 or plug-in 258, a file system access request. The file system access request can include, but is not limited to, a copy request, an open request, a move request, a delete request, and the like.

[0104] The filter driver 254 or plug-in 258 analyzes the meta-data associated with the digital asset. The analysis includes, processing the meta-data to determine (step 1320) which taxonomy tags 430 are associated with the digital asset. The analysis also includes evaluating (step 1330) the rules that are associated with the applied taxonomy tags 430. For example, if a digital asset was tagged as Finance/Budget, the associated rules can be to restrict access to only all the executives and John, who is a consultant, when he is accessing the digital asset from a computer located at the offices of the corporation. The filter driver 254 intercepts the file system request for access and ensures that that each of the conditions is satisfied. If each of the conditions is satisfied, the requested access is allowed (step 1340). Although described from the perspective of the client 110, it should be understood the server 150 can also perform the described method.

[0105] It is also desirable to prevent unauthorized access to digital assets by trying to circumvent the rules and categorizations associated with the digital assets. To that end, the invention propagates the meta-data with the digital asset. For example, if a digital asset is attached to an email the meta-data is also attached to the email. Similarly, if a digital asset is copied to a storage device, the meta-data associated with the digital asset is copied as well.

[0106] FIG. 14 depicts an embodiment of a method of propagating the meta-data with a digital asset. The method includes generating (step 1410) a meta-data set for a digital asset, associating (step 1420) the meta-data set with the digital asset, and transferring (step 1430) the meta-data set with the digital asset.



[0107] The various methods of generating a meta-data set for a digital asset have been described above and will not be repeated here. The meta-data set can have characteristics of the digital asset. For example, if the digital asset is a word file, the meta-data set can have certain properties of a word file as well. In one embodiment, the meta-data set is a hidden file.

[0108] The associated meta-data set is transferred (step 1430) with the digital asset. Transferring can include, but is not limited to, copying, renaming, deleting, moving, emailing, and the like. In the case a digital asset is transferred as an attachment to an email, the meta-data can be transferred using certain aspects of the email. It is known, when an email is transmitted to a recipient the email format is defined by standards from the IETF such as RFC 822 or the newer RFC 2822, the entire contents of which are herein incorporated by reference. These standards provide for fields in the email header such as comments, keywords and an optional-field. The meta-data can be placed into these fields using the plug-in 258 of the client software 212 or by a network filter driver 254 (not shown) that is located in the network driver stack of the client 110. By sending the meta-data with the digital asset, the meta-data is received at the same time as the digital asset.

[0109] In certain instances, it is desirable to suspend the creation of meta-data for digital asset. For example, during the installation of other software applications. Typically, when installing software, a program is executing commands that will cause folders to be created. The end-user could be bombarded with requests for categorization for folders. Because of this, there is a special command that can be executed by the end-user of the client 110 that informs the client software 212 to suspend its operation.

[0110] Referring back to FIG. 4, the adaptive feature of the adaptive taxonomy is described. One aspect of the invention is the ability of end-users and administrators to create aliases 450 to taxonomy tags 430 to provide an adaptive taxonomy 440. The aliases can be available only the end-user of the client 110 or available globally to all clients 110. Additionally, an alias can be promoted to taxonomy tag 440. The invention also provides a means to set a policy describing which end-users can create aliases 450. As shown in FIG. 4, the alias 450 labeled "INVENTION" refers to the taxonomy tag 440 labeled "DISCLOSURE." Similarly, the alias 450 labeled "CV" is

an alias for the taxonomy tag 440 "RESUME." Each alias inherits each of the rules and categorizations of the taxonomy tag 440 to which it refers.

[0111] With reference to FIG. 15, a method of creating an alias for a taxonomy tag is shown and described. The method includes creating (step 1510) a digital asset, presenting (step 1520) all or a portion of the taxonomy tags 440 to the end-user, providing (step 1530) a graphical user interface to the end-user if the end-user performs a specified action, and creating (step 1540) an alias using the graphical user interface presented to the end-user.

[0112] As previously explained the end-user can apply a presented taxonomy tag to a digital asset before the digital asset is saved to the file system of the client 110 or after the digital asset is stored at the client 110. The taxonomy tag 440 categorizes the digital asset and typically includes at least one rule for the digital asset.

[0113] The end-user can specifically request the formation of an alias by performing a specified action. The action can be, but is not limited to, selecting a button presented with the taxonomy tags 440 or not selecting one of the presented taxonomy tags. Once the client software 212 determines that the end-user wants to create an alias 450, a graphical user interface is presented to the end-user that allows the end-user to create the alias 450. The end-user supplies a required set of characteristics of the alias. For example, to which taxonomy tag 440 the alias 450 refers and an associated policy. The policy can be the same policy as the taxonomy tag 440 or a more restrictive policy.

[0114] The following example is designed to illustrate the adaptive taxonomy features of the invention and is not intended to limit the invention. Referring to FIG. 4, the tag 430 labeled "Resume" that is used to categorize resume information in the HR department. A policy is associated with the Resume tag 440 that implements a first policy "A". After policy A is in place and another end-user of the HR department decides that the department needs a new tag called "CV". The end-user is presented with a graphical user interface that requires the end-user to provide a taxonomy tag 440 to which the alias 450 CV is linked (i.e., Resume), the reason for creating the alias 450, and whether a more restrictive policy "B" should be applied to digital assets tagged with the CV alias 450.



[0115] The alias 450 is able to be used by the end-user locally. However, it may be desirable to allow other end-users to use the same alias 450. To the end, the alias 450 is transmitted to the server 150 for review by an administrator.

[0116] As previously stated, the server software 312 provides certain features of the invention alone and in combination with the client software 212. Examples of features provided by the server software 312 include, promoting an alias 450 to a tag 430 and modifying the adaptive taxonomy 400, performing unions of meta-data sets for digital assets, and locating digital assets in the distributed computing environment. Each of these features is discussed below in more detail.

[0117] An administration reviews the aliases that have been created by the end-users of the client 110 on a periodic basis. Continuing with the above example, if the administrator agrees with the request to make an alias 450 called "CV" the administrator modifies the taxonomy 400 to include the alias 450 CV using the server software 312. As previously mentioned, the alias 450 can have the same policies as the Resume taxonomy tag 440 or a more restrictive policy. The updated taxonomy 400 is transferred to each client 110 the next time the client connects to the server 150.

[0118] Alternatively, administrator can deny the alias 450 CV. As a result, the alias 450 CV is only available local to the end-user of the client 110 that created the alias. Said another way, the alias 450 CV is not published to the other clients 110. The end-user can also remove local aliases as needed. As such, the digital asset is then tagged with the taxonomy tag that the alias referred to prior to deletion.

[0119] Additionally, the administrator can "promote" an alias 450 to a taxonomy tag 430. In essence, a promotion from an alias 450 to a taxonomy tag 430 has the same effect as adding a new tag 430 to the adaptive taxonomy 400. Continuing with the with above example, if it is later decided that a different policy should be applied to digital assets categorized as a CV versus those categorized as Resumes, the administrator can promote the alias 450 labeled as CV to a taxonomy tag 430 and revise the associated policy for the CV taxonomy tag 430.

[0120] It is conceivable that the same digital asset exists on multiple clients 110. Each of the end-users can apply a different taxonomy tag 430 to the digital asset. If

the digital asset is a records managed asset, a copy of each of the meta-data sets associated with the digital asset are stored at the server 150. Having different policy information with the same digital asset may allow for circumvention of the desired result of the present invention. To that end, a method of unionizing the meta-data sets for the digital asset is performed by, in one embodiment, the server software 312.

**[0121]** One embodiment of a method 1600 of unionizing the associated meta-data tags is shown and described with reference to FIG. 16. The method includes receiving (step 1610) a first meta-data set for a digital asset from a first client 110, receiving (step 1620) a second meta-data set for the same digital asset from a second client 110, and selecting (step 1630) one of the categorizations of the digital asset of as the active categorization. Although the other categorizations are present in the meta-data, only the active categorization and its associated policies are enforced with respect to the digital asset.

**[0122]** Various methods are used to determine which categorization to select. In one embodiment, the more restrictive categorization is selected. For example, the categorization that allows the fewest end-uses to access the digital asset is selected. Other examples include selecting the categorization that allows the largest number of end-users access to the digital asset, selecting the categorization that permits the fewest number of actions to be performed on the file, selecting the categorization that allows the largest number of actions to be performed on the file, selecting the categorization having the earliest association date.

**[0123]** In one embodiment, the following method is used to determine which categorization to select. First a comparison between the retention policies is performed and the categorization having the longer retention policy is applied. If the retention policies are equal, then a comparison of the expiration policies is performed. Again, the categorization having the longer expiration policy is applied. If expiration policies are equal, then the end-user is queried to provide a ranking to each policy to resolve the conflict. In one embodiment, the invention includes a policy analysis engine that analyzes the policies when they are created. In the case of competing policies, the end-user is queried to rank the competing policies to the resolution of competing policy as applied to the digital assets occurs automatically.



[0124] Each of the categorization remains with the meta-data set for the digital asset. The not active categorization is not removed from the meta-data set. The reason for this is that different groups or end-users within an organization can view the value of a digital asset. For example, the legal department can view an offer letter as a contract, human resources can view the same offer letter as a salary benchmark, and manufacturing can view the offer letter as just a letter. This information is included as part of the audit information of the meta-data set. The below example illustrates certain aspects of the invention.

[0125] Once the server software 312 identifies that the meta-data sets are for the same digital asset the process of unionizing the meta-data set for the digital asset is executed. For purposes of this example, assume that an end-user of a first client categorized a digital asset "General Corporate" and another end-user categorizes the same digital asset as "Budget". Both categorizations are correct, but one is more correct. The server software 312 determines which categorization is stricter and selects that categorization as the active categorization.

[0126] The server 150 saves the information for each of the meta-data sets in a master meta-data set for the digital asset. The master meta-data set includes the information from each of the meta-data sets. The master meta-data set becomes the meta-data set for the digital asset and is communicated to each client 110 the next time the client establishes a connection with the server 150.

[0127] Although described with reference to the server 150, it should be understood that the client 110 is also capable of unionizing the associated meta-files. The functionality is provided and used by the client software 212. For example when a first file and a second file are each copied and pasted into a third file, the client software 212 performs a union of the first files meta-data the second files meta-data to generate a master meta-data set for the third file.

[0128] Another feature provided by the invention is the ability to located and freeze the state of digital assets with in the distributed computing environment 100. In one embodiment, this feature is accomplished by the cooperation of the server software 312 and the client software 212. One method of capturing a set of digital assets is shown and described with reference to FIG. 17.

[0129] In one embodiment, the method includes receiving (step 1710) by a client 110 an instruction from the server 150 to copy specific digital assets identified by the meta-data for the digital asset, copying (step 1720) the identified digital assets, associating (step 1730) a respective audit trail to each of the respective copied digital asset, and transmitting (step 1740) the digital assets and their associated audit trails to the server 150.

[0130] To illustrate some of the features of the invention, the following example is provided. The example illustrates how an administrator of server 150 uses the software system of the invention to select a set of digital assets in the distributed computing environment 100 to be frozen and produced in litigation. The result of this operation is a report of digital assets of interest along with the locations of the information.

[0131] Each client 110 periodically checks with server 150 for issued instruction. If an instruction exists, the client 110 receives the instructions. The periodicity can vary and can also be overridden. For example, the end-user of a client 110 can issue a connection request by performing an operation using the client 110. In response to receiving the instructions, the client software 212 analyzes the instructions and begins their execution. In this example, assume the server 150 instructs the client 110 to prevent modification (i.e., freeze) and generate copies of indicated digital assets and their associated meta-data, which includes the audit history of the digital asset. After the server software 212 copies a respective digital asset, that digital asset is released from the hold state so that the end-user of the client can access the digital asset. The client 110 transmits the copy of the digital asset and meta-data to the server 150.

[0132] Each client 110 of the distributed computing environment 100 performs the copy operating in parallel with the other clients 110. The server software 312 includes functionality to provide a status report displaying the number or percentage of clients 110 that received the instruction, the number of clients 110 still to receive the instruction, and the number of clients 110 that have completed the copy and transmission operations. It should be understood other progress metrics can be included in the reporting functionality of the invention.

[0133] The previously described embodiments may be implemented as a method, apparatus or article of manufacture using programming and/or engineering techniques



to produce software, firmware, hardware, or any combination thereof. The term "article of manufacture" as used herein is intended to encompass code or logic accessible from and embedded in one or more computer-readable devices, firmware, programmable logic, memory devices (e.g., EEPROMs, ROMs, PROMs, RAMs, SRAMs, etc.), hardware (e.g., integrated circuit chip, Field Programmable Gate Array (FPGA), Application Specific Integrated Circuit (ASIC), etc.), electronic devices, a computer readable non-volatile storage unit (e.g., CD-ROM, floppy disk, hard disk drive, etc.), a file server providing access to the programs via a network transmission line, wireless transmission media, signals propagating through space, radio waves, infrared signals, etc. The article of manufacture includes hardware logic as well as software or programmable code embedded in a computer readable medium that is executed by a processor. Of course, those skilled in the art will recognize that many modifications may be made to this configuration without departing from the scope of the present invention.

[0134] While the invention has been shown and described with reference to specific preferred embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention as defined by the following claims.

[0135] What is claimed is:

**CLAIMS**

1. A method of managing at least one digital asset in a distributed computing environment, the method comprising:
  - (a) generating, by a client, an asset identification tag for a digital asset stored at the client;
  - (b) associating the asset identification tag with the digital asset; and
  - (c) storing the asset identification tag at the client.
2. The method of claim 1 wherein storing comprises storing the asset identification tag separate from the digital asset.
3. The method of claim 1 further comprising transmitting a copy of the asset identification tag to a central computing device thereby notifying the central computing device of the existence of the digital asset.
4. The method of claim 1 wherein step (a) occurs after the digital asset is stored.
5. The method of claim 1 wherein step (a) occurs after the digital asset is accessed by the end-user of the client.
6. The method of claim 1 wherein the asset identification tag comprise categorization information about the digital asset.
7. The method of claim 6 wherein the categorization information is supplied by an end-user through a graphical user interface.
8. The method of claim 6 wherein the categorization information is applied based on a predetermined taxonomy.
9. The method of claim 6 wherein the categorization information is applied based on a role of a sender of an email.
10. The method of claim 6 wherein the categorization information is applied based on a role of a recipient of an email.
11. The method of claim 6 wherein the categorization is applied based on an attachment of an email.



12. The method of claim 6 wherein the categorization information is applied automatically based on the storage location of the digital asset.
13. The method of claim 12 wherein the location is another storage device.
14. The method of claim 6 wherein the categorization information comprises policy information that is applied to the digital asset.
15. The method of claim 14 wherein the policy information is supplied by an end-user through a graphical user interface.
16. The method of claim 14 wherein the policy information is applied based on a predetermined taxonomy.
17. The method of claim 14 wherein the policy information is applied based on the storage location of the digital asset.
18. The method of claim 1 wherein the asset identification tag comprises the result of a hash function applied to the digital asset.
19. The method of claim 1 further comprising sending an instruction to the client by the central computing device when the copy of the asset identification tag is received and processed by the central computing device.
20. The method of claim 19 wherein the instruction causes the client to perform a specific action on the digital asset.
21. The method of claim 1 further comprising updating the asset identification tag when the end-user accesses or manipulates the digital asset.
22. The method of claim 21 further comprising transmitting the updated asset identification tag to the server.
23. The method of claim 1 further comprising suspending steps (a), (b), and (c) when a software application is installed on the client.
24. A system for managing at least one digital asset in a distributed computing environment the system comprising:
  - (a) a server connected to a network; and
  - (b) a client in communication with the server, the client generating an asset identification tag for a digital asset stored on the client,

associating the asset identification tag with the digital asset, and  
storing the asset identification tag at the client.

25. The system of claim 24 further comprising a storage medium in communication with the client, the storage medium storing the asset identification tag separate from the associated digital asset.
26. The system of claim 24 wherein the client comprises a transmitter for transmitting a copy of the asset identification tag to the server, the server storing the digital asset.
27. The system of claim 24 further comprising a database in communication with the client, the database storing the asset identification tag.
28. The system of claim 24 further comprising a database in communication with server, the database storing the asset identification tag received by the server.
29. The system of claim 24 wherein the client generates the asset identification tag after the digital asset is stored.
30. The system of claim 24 wherein the client generates the asset identification tag after the digital asset is accessed.
31. The system of claim 24 wherein the asset identification tag comprises categorization information.
32. The system of claim 31 wherein the categorization information is applied based on a predetermined taxonomy.
33. The system of claim 31 wherein the categorization information is applied based on the role of the sender of an email.
34. The system of claim 31 wherein the categorization information is applied based on the role of the receipt of the an email.
35. The system of claim 31 wherein the categorization information is applied based on the categorization of an email attachment.
36. The system of claim 31 wherein the categorization information is applied automatically based on the storage location of the digital asset.



37. The system of claim 31 wherein the categorization information comprises policy information applied to the digital asset.
38. The system of claim 37 further comprising a graphical user interface displayed to the end-user, the graphical user interface configured to permit policy information to be applied.
39. The system of claim 37 wherein the policy information is applied based on the storage location of the digital asset.
40. The system of claim 24 wherein the client further comprises a receiver configured to receive an instruction from the server.
41. The system of claim 40 wherein the instruction causes the client to perform a specified action on the digital asset.
42. The system of claim 24 wherein the client updates the asset identification tag when the end-user accesses or manipulates the digital asset.
43. The system of claim 42 wherein the client comprises a transmitter for transmitting the updated asset identification tag to the server.
44. The system of claim 24 wherein the asset identification tag comprises the result of a hash function applied to the digital asset.
45. The system of claim 24 further comprising a graphical user interface displayed to the client to allow an end-user of the client to suspend the operation of the client during the installation of a software program.
46. A computer readable medium having executable instructions thereon to cause the management of at least one digital asset in a distributed computing environment, the computer readable medium comprising:
  - (a) instructions to generate, an asset identification tag for a digital asset stored at the client;
  - (b) instructions to associate the asset identification tag with the digital asset; and
  - (c) instructions to store the asset identification tag at the client.

47. The computer readable medium of claim 46 wherein the instructions to store comprise instructions to store the asset identification tag separate from the digital asset.
48. The computer readable medium of claim 46 further comprising instructions to transmit a copy of the asset identification tag to a central computing device thereby notifying the central computing device of the existence of the digital asset.
49. The computer readable medium of claim 46 wherein step (a) is executed after the digital asset is stored.
50. The computer readable medium of claim 46 wherein step (a) is executed after the digital asset is accessed by the end-user of the client.
51. The computer readable medium of claim 46 wherein the instructions to generate an asset identification tag comprise instructions to apply categorization information about the digital asset.
52. The computer readable medium of claim 51 wherein the categorization information is supplied by an end-user through a graphical user interface.
53. The computer readable medium of claim 51 wherein the categorization information is applied based on a predetermined taxonomy.
54. The computer readable medium of claim 51 wherein the categorization information is applied based on a role of a sender of an email.
55. The computer readable medium of claim 51 wherein the categorization information is applied based on a role of a recipient of an email.
56. The computer readable medium of claim 51 wherein the categorization is applied based on an attachment of an email.
57. The computer readable medium of claim 51 wherein the categorization information is applied automatically based on the storage location of the digital asset.
58. The computer readable medium of claim 57 wherein the location is another storage device.



59. The computer readable medium of claim 51 wherein the categorization information comprises policy information that is applied to the digital asset.
60. The computer readable medium of claim 59 wherein the policy information is supplied by an end-user through a graphical user interface.
61. The computer readable medium of claim 59 wherein the policy information is applied based on a predetermined taxonomy.
62. The computer readable medium of claim 59 wherein the policy information is applied based on the storage location of the digital asset.
63. The computer readable medium of claim 46 wherein the instructions to generate the asset identification tag comprises instructions to generate the result of a hash function to apply to the digital asset.
64. The computer readable medium of claim 51 further comprising instructions to send an instruction to the client by the central computing device when the copy of the asset identification tag is received and processed by the central computing device.
65. The computer readable medium of claim 64 wherein the instruction causes the client to perform a specific action on the digital asset.
66. The computer readable medium of claim 46 further comprising instructions to update the asset identification tag when the end-user accesses or manipulates the digital asset.
67. The computer readable medium of claim 66 further comprising instructions to transmit the updated asset identification tag to the server.
68. The computer readable medium of claim 46 further comprising instructions to suspend steps (a), (b), and (c) when a software application is installed on the client.
69. A method of providing access to a digital asset of a distributed computing environment, the environment having a central computing device that stores an asset identification tag for at least one digital asset and one or more clients that locally stores respective digital assets, the method comprising:

- (a) receiving a request from a user for access to a digital asset;
- (b) determining a categorization associated with the digital asset;
- (c) evaluating a rule associated with the categorization of the digital asset to generate a result; and
- (d) providing access to the digital asset when the result indicates that the end-user is allowed to access the digital asset.

70. The method of claim 69 wherein at least one of steps (a), (b), and (c) is performed at the client.
71. The method of claim 69 wherein at least one of steps (a), (b) and (c) is performed by the central computing device.
72. The method of claim 69 further comprising receiving, at a client, the rule associated with the categorization from the central computing device.
73. The method of claim 69 wherein step (a) is performed by a filter driver.
74. The method of claim 69 wherein providing access comprises performing an operation on the digital asset selected from the group consisting of: reading, writing, deleting, copying, moving, transmitting, transferring, and renaming.
75. A system for providing access to digital assets of a distributed computing environment, the environment having a central computing device that stores an asset identification tag for at least one digital asset and one or more clients that locally stores respective digital assets, the system comprising:
- (a) a computing device that receives a request from a user for access to digital assets, determines a categorization associated with the digital asset, evaluates a rule associated with the categorization of the digital asset to generate a result, and provide access to the digital asset when the result indicates that the end-user is allowed to access the digital asset.
76. The system of claim 75 wherein the computing device is the client of the distributed computing environment.



77. The system of claim 75 wherein the computing device is the server of the distributed computing environment.
78. The system of claim 75 wherein the server of the distributed computing environment transmits the rule associated with the categorization to the computing device.
79. The system of claim 75 wherein the computing device comprises a filter driver.
80. The system of claim 75 wherein the computing device provides access by performing an operation on the digital asset selected from the group consisting of: reading, writing, deleting, copying, moving, transmitting, transferring, and renaming
81. A computer readable medium having executable instructions thereon to provide access to a digital asset of a distributed computing environment, the environment having a central computing device that stores an asset identification tag for at least one digital asset and one or more clients that locally stores respective digital assets, the computer readable medium comprising:
- (a) instructions to receive a request from a user for access to a digital asset;
  - (b) instructions to determine a categorization associated with the digital asset;
  - (c) instructions to evaluate a rule associated with the categorization of the digital asset to generate a result; and
  - (d) instructions to provide access to the digital asset when the result indicates that the end-user is allowed to access the digital asset.
82. The computer readable medium of claim 81 wherein the execution of steps (a), (b), and (c) is performed at a client of the distributed computing environment.
83. The computer readable medium of claim 81 wherein the execution of steps (a), (b), and (c) is performed at the server of the distributed computing environment.

84. The computer readable medium of claim 81 further comprising instructions to receive, at a client of the distributed computing environment, the rule associated with the categorization from the central computing device.
85. The computer readable medium of claim 81 wherein the instructions to receive a request is performed by a filter driver.
86. The computer readable medium of claim 81 wherein the instructions to provide access comprises instructions to perform an operation on the digital asset selected from the group consisting of: reading, writing, deleting, copying, moving, transmitting, transferring, and renaming.
87. A method of preventing the restoration of a digital asset that has been removed from a distributed computing environment, the environment having a central computing device that stores meta-data for at least one digital asset and one or more clients that locally stores respective digital assets, the method comprising:
- (a) receiving, from a client, meta-data associated with a digital asset created by the client;
  - (b) comparing at least a portion of the received meta-data with a list of known meta-data to generate a result; and
  - (c) transmitting a command to remove the digital asset associated with the received meta-data when the result indicates that the digital asset is expired.
88. The method of claim 87 further comprising establishing a first connection to receive the meta-data.
89. The method of claim 87 further comprising establishing a second connection to transmit the command to remove the digital asset.
90. The method of claim 87 wherein the portion of the meta-data comprises an asset identification tag.
91. The method of claim 90 wherein the asset identification tag comprises the result of a hash function applied to the digital asset.



92. The method of claim 87 wherein the receiving is performed by the central computing device.
93. The method of claim 87 wherein the list of known meta-data comprises a database.
94. A system for preventing the restoration of a digital asset that has been removed from a distributed computing environment, the environment having a central computing device that stores meta-data for at least one digital asset and one or more clients that locally stores respective digital assets, the system comprising:
- (a) a client in communication with a network, the client receiving meta-data associated with a digital asset created by the client and transmitting the meta-data to the network; and
  - (b) a server connected to a network, the server receiving the meta-data and comparing at least a portion of the received meta-data with a list of known meta-data to generate a result, and transmitting a command to remove the digital asset associated with the received meta-data when the result indicates that the digital asset is expired.
95. The system of claim 94 wherein the client establishes a first connection to receive the meta-data.
96. The system of claim 94 wherein the server establishes a second connection to transmit the command to remove the digital asset.
97. The system of claim 94 wherein the portion of the meta-data comprises an asset identification tag.
98. The system of claim 97 wherein the asset identification tag comprises the result of a hash function applied to the digital asset.
99. The system of claim 94 further comprising a database in communication the server that stores the list of known meta-data.
100. A computer readable medium having executable instructions thereon to prevent the restoration of a digital asset that has been removed from a distributed computing environment, the environment having a central

computing device that stores meta-data for at least one digital asset and one or more clients that locally stores respective digital assets, the computer readable medium comprising:

- (a) instructions to receive, from a client, meta-data associated with a digital asset created by the client;
- (b) instructions to compare at least a portion of the received meta-data with a list of known meta-data to generate a result; and
- (c) instruction to transmit a command to remove the digital asset associated with the received meta-data when the result indicates that the digital asset is expired.

- 101. The computer readable medium of claim 100 further comprising instructions to establish a first connection to receive the meta-data.
- 102. The computer readable medium of claim 100 further comprising instructions to establish a second connection to transmit the command to remove the digital asset.
- 103. The computer readable medium of claim 100 wherein the portion of the meta-data comprises an asset identification tag.
- 104. The computer readable medium of claim 103 wherein the asset identification tag comprises the result of a hash function applied to the digital asset.
- 105. The computer readable medium of claim 100 wherein the instructions to receive are executed by the central computing device.
- 106. The computer readable medium of claim 100 wherein the list of known meta-data comprises a database.
- 107. A method of locating a digital asset in a distributed computing environment, the environment having a central computing device that stores meta-data for at least one digital asset and one or more clients that locally stores respective digital assets, the method comprising:
  - (a) receiving a search command from an end-user of the client;



- (b) identifying at least one taxonomy tag associated with a digital asset stored by the client; and
- (c) displaying at least one folder to the end-user, the at least one folder having a label associated with the at least one identified taxonomy tag.

108. The method of claim 107 further comprising identifying at least one keyword by processing the digital assets stored by the client.
109. The method of claim 108 further comprising displaying at least one additional folder containing a reference to a digital asset including the at least one keyword.
110. The method of claim 108 further comprising displaying at least one folder having a label comprising the identified keyword.
111. The method of claim 107 wherein the at least one folder is displayed graphically to the end-user.
112. The method of claim 107 further comprising intercepting the search command by a filter driver.
113. The method of claim 107 further comprising intercepting the search command by a plug-in module.
114. A system for locating a digital asset in a distributed computing environment, the environment having a central computing device that stores meta-data for at least one digital asset and one or more clients that locally stores respective digital assets, the system comprising:
- a computing device that receives a search command from an end-user of the computing device, identifies at least one taxonomy tag associated with a digital asset stored by the client, and displays at least one folder to the end-user, the at least one folder having a label associated with the at least one identified taxonomy tag.
115. The system of claim 114 wherein the computing device identifies at least one keyword by processing the digital assets stored by the computing device.

116. The system of claim 115 wherein the computing device displays at least one additional folder containing a reference to a digital asset including the at least one keyword.
117. The system of claim 115 wherein the computing device displays at least one folder having a label comprising the identified keyword.
118. The system of claim 114 wherein the at least one folder is displayed graphically to the end-user.
119. The system of claim 114 wherein the computing device comprises a filter driver that intercepts the search command.
120. The system of claim 114 wherein the computing device comprises a plug-in module that intercepts the search command.
121. A computer readable medium having executable instructions thereon to locate a digital asset in a distributed computing environment, the environment having a central computing device that stores meta-data for at least one digital asset and one or more clients that locally stores respective digital assets, the computer readable medium comprising:
- (a) instructions receive a search command from an end-user of the client
  - (b) instructions to identify at least one taxonomy tag associated with a digital asset stored by the client; and
  - (c) instruction to display at least one folder to the end-user, the at least one folder having a label associated with the at least one identified taxonomy tag.
122. The computer readable medium of claim 121 further comprising instructions to identify at least one keyword by processing the digital assets stored by the client.
123. The computer readable medium of claim 122 further comprising instructions to display at least one additional folder containing a reference to a digital asset including the at least one keyword.



124. The computer readable medium of claim 122 further comprising instructions to displays at least one folder having a label comprising the identified keyword.
125. The computer readable medium of claim 121 wherein the at least one folder is displayed graphically to the end-user.
126. The computer readable medium of claim 121 further comprising instructions to cause a filter driver to intercept the search command.
127. The computer readable medium of claim 121 further comprising instructions to cause a plug-in module to intercept the search command.
128. A method of tracking the number of copies of a digital asset in a distributed computing environment, the environment having a central computing device that stores meta-data for at least one digital asset and one or more clients that locally stores respective digital assets and their associated meta-data, the method comprising:
- (a) determining, by a client, if a copy of an original digital asset is created by the client;
  - (b) generating, by the client, a meta-data entry indicating that a copy of the original digital asset was created by the client when the determination indicates that a copy was created; and
  - (c) updating the stored meta-data for the original digital asset.
129. The method of claim 128 wherein the determining comprises comparing an asset identification tag created for the copy with a set of known asset identification tags.
130. The method of claim 128 wherein the generating comprising generating a meta-data entry indicating that a printed copy of the original digital asset was created.
131. The method of claim 128 further comprising applying categorization information associated with the original digital asset to the copy of the original digital asset.

132. The method of claim 131 wherein the applying comprises performing a union of categorization information when more than one categorization of the original digital asset is present.
133. The method of claim 128 wherein the updating comprises updating a local copy of the meta-data for the original digital asset.
134. The method of claim 133 further comprising updating a remote copy of the meta-data of the original digital asset.
135. A system for tracking the number of copies of a digital asset in a distributed computing environment, the environment having a central computing device that stores meta-data for at least one digital asset and one or more clients that locally stores respective digital assets and their associated meta-data, the system comprising:
- (a) a server connected to a network; and
  - (b) a client in communication with the server, the client determining if a copy of an original digital asset is created by the client, generating a meta-data entry indicating that a copy of the original digital asset was created by the client when the determination indicates that a copy was created, updating the stored meta-data for the original digital asset.
136. The system of claim 135 wherein the client comprises a comparator for comparing an asset identification tag created for the copy with a set of known asset identification tags.
137. The system of claim 135 wherein the client generates a meta-data entry indicating that a printed copy of the original digital asset was created.
138. The system of claim 135 wherein the client applies categorization information associated with the original digital asset to the copy of the original digital asset
139. The system of claim 138 wherein the client performs a union of categorization information when more than one categorization of the original digital asset is present



140. The system of claim 135 wherein the client updates a local copy of the meta-data for the original digital asset
141. The system of claim 140 wherein the client updates a remote copy of the meta-data of the original digital asset stored on the server.
142. A computer readable medium having executable instructions thereon to track the number of copies of a digital asset in a distributed computing environment, the environment having a central computing device that stores meta-data for at least one digital asset and one or more clients that locally stores respective digital assets and their associated meta-data, the computer readable medium comprising:
- (a) instructions to determine, by a client, if a copy of an original digital asset is created by the client;
  - (b) generate, by the client, a meta-data entry indicating that a copy of the original digital asset was created by the client when the determination indicates that a copy was created; and
  - (c) update the stored meta-data for the original digital asset.
143. The computer readable medium of claim 142 wherein the instructions to determine comprise instructions to compare an asset identification tag created for the copy with a set of known asset identification tags.
144. The computer readable medium of claim 142 wherein the instructions to generate comprise instructions to generate a meta-data entry indicating that a printed copy of the original digital asset was created.
145. The computer readable medium of claim 142 further comprising instructions to apply categorization information associated with the original digital asset to the copy of the original digital asset.
146. The computer readable medium of claim 145 wherein the instructions to apply comprise instructions to perform a union of categorization information when more than one categorization of the original digital asset is present.

147. The computer readable medium of claim 145 wherein the instructions to update comprise instructions to update a local copy of the meta-data for the original digital asset.
148. The computer readable medium of claim 146 further comprising instructions to update a remote copy of the meta-data of the original digital asset.
149. A method of capturing documents in a distributed computing environment at a central location based on specified criteria, the environment having a central computing device that stores an asset identification tag for at least one digital asset and one or more clients that locally stores respective digital assets, the method comprising:
- (a) receiving, at a client, an instruction from the central computing device, to copy a digital asset, the instruction including an asset identification tag;
  - (b) copying the digital asset identified by the asset identification tag;
  - (c) associating an audit trail with digital asset in response to the instruction from the central computing device; and
  - (d) transmitting the copy of the digital asset and the associated audit trail of the digital asset to another location.
150. The method of claim 149 further comprising checking with the central computing device, by the client, for the instruction.
151. The method of claim 150 wherein the checking is performed on a periodic basis.
152. The method of claim 149 further comprising preventing access to the digital asset in response to receiving the instruction from the central computing device.
153. The method of claim 149 further comprising checking with the central computing device when the end-user of the client performs a specified action.



154. The method of claim 149 further comprising transmitting the instruction to the client when an end-user of the central computing device performs a specified action.
155. The method of claim 149 further comprising generating a report, by the central computing device, the report indicating an amount of copies of digital assets that have been transmitted to the another location.
156. A system for capturing documents in a distributed computing environment at a central location based on specified criteria, the environment having a central computing device that stores an asset identification tag for at least one digital asset and one or more clients that locally stores respective digital assets, the system comprising:
- (a) a central computing device to a network; and
  - (b) a client in communication with the server, the client receiving an instruction from the central computing device to copy a digital asset, the instruction including an asset identification tag, copying the digital asset identified by the asset identification tag, associating an audit trail with the digital asset in response to the instruction from the central computing device, and transmitting the copy of the digital asset and the associated audit trail of the digital asset to another location.
157. The system of claim 156 wherein the client checks with the central computing device for the instruction.
158. The system of claim 157 wherein the checking is performed on a periodic basis.
159. The system of claim 156 wherein the client prevents access to the digital asset in response to the instruction from the central computing device.
160. The system of claim 156 wherein the client checks with central computing device when the end-user of the client performs a specified action.
161. The system of claim 156 wherein the central computing device transmits the instruction to the client when an end-user of the central computing device performs a specified action.

162. The system of claim 156 wherein the central computing device generates a report, the report indicating an amount of copies of digital assets that have been transmitted to the another location
163. A computer readable medium having executable instructions thereon to capture documents in a distributed computing environment at a central location based on specified criteria, the environment having a central computing device that stores an asset identification tag for at least one digital asset and one or more clients that locally stores respective digital asset, the computer readable medium comprising:
- (a) instructions to receive, at a client, an instruction from the central computing device, to copy a digital asset, the instruction including an asset identification tag;
  - (b) instructions to copy the digital asset identified by the asset identification tag;
  - (c) instructions to associate an audit trail with digital asset in response to the instruction from the central computing device; and
  - (d) instructions to transmit the copy of the digital asset and the associated audit trail of the digital asset to another location.
164. The computer readable medium of claim 163 further comprising instructions to check with the central computing device, by the client, for the instruction.
165. The computer readable medium of claim 164 wherein the instructions to check is performed on a periodic basis.
166. The computer readable medium of claim 165 further comprising instructions to prevent access to the digital asset in response to receiving the instruction from the central computing device.
167. The computer readable medium of claim 165 further comprising instructions to check with the central computing device when the end-user of the client performs a specified action.



168. The computer readable medium of claim 165 further comprising instructions to transmit the instruction to the client when an end-user of the central computing device performs a specified action.
169. The computer readable medium of claim 165 further comprising instructions to generate a report, by the central computing device, the report indicating an amount of copies of digital assets that have been transmitted to the another location.
170. A method of copying digital assets having an associated expiration date in a distributed computing environment, the method comprising:
- (a) providing a date range;
  - (b) enumerating digital assets as file system elements having an associated expiration date in the provided date range; and
  - (c) storing the enumerated digital assets on a storage medium that is removable from the distributed computing environment.
171. The method of claim 170 wherein providing a date range comprises providing the date range using a system call.
172. The method of claim 170 further comprising copying the enumerated digital asset to another storage medium.
173. The method of claim 170 wherein the providing a date range comprises providing a desired expiration date range.
174. The method of claim 170 wherein the providing a date range comprises providing a flag in addition to a desired expiration date range, the flag indicating the opposite of the desired expiration date range.
175. The method of claim 170 wherein at least step (b) is performed by a file system filter driver.
176. The method of claim 170 wherein at least step (b) is performed by a plug-in module.
177. A system for coping digital assets having an associated expiration date in a distributed computing environment, the system comprising:

(a) a computing device in communication with a network of the distributed computing environment, the computing device receiving a date range, enumerating digital assets as file system elements having an associated expiration date in the provided date range, and storing the enumerated digital assets on a storage medium that is removable from the distributed computing environment e.

178. The system of claim 177 wherein the date range is provided using a system call.
179. The system of claim 177 wherein the enumerated digital assets are copied to another storage medium.
180. The system of claim 177 wherein the date range indicates a desired expiration date range.
181. The system of claim 177 wherein the date range include a flag in addition to a desired date range , the flag indicating the opposite of the desired date range.
182. The system of claim 177 wherein the computing device comprises a filter driver.
183. The system of claim 177 wherein the computing device comprises a plug-in module.
184. A computer readable medium having executable instructions thereon to copy digital assets having an associated date expiration date in a distributed computing environment., the computer readable medium comprising:
- (a) instructions to receive a date range;
  - (b) instructions to enumerating digital assets as file system elements having an associated expiration date in the provided date range; and
  - (c) instruction to store the enumerated digital assets on a storage medium that is removable from the distributed computing environment.



185. The computer readable medium of claim 184 wherein the instructions to receive a date range comprise instructions to receive the date range using a system call.
186. The computer readable medium of claim 184 further comprising instructions to copy the enumerated digital asset to another storage medium.
187. The computer readable medium of claim 184 wherein the instructions to provide a date range comprises instructions to provide a desired expiration date range
188. The computer readable medium of claim 184 wherein the instructions to provide a date range comprise instructions to provide a flag in addition to a desired expiration date range, the flag indicating the opposite of the desired expiration date range.
189. A method of preventing access to an expired digital asset, the method comprising:
- (a) encrypting a digital asset using an encryption key, the digital asset associated with a tag specifying a time after which the digital asset should be unavailable, the encryption key having a predetermined expiration time; and
  - (b) destroying the encryption key after the predetermined time.
190. The method of claim 189 further comprising marking the digital asset for expiration prior to encrypting the digital asset.
191. The method of claim 189 further comprising:
- copying the encrypted digital asset from a file system to another storage medium;
  - copying the encryption key to the another storage medium; and
  - wherein the destroying comprises, destroying the encryption key on the another storage medium after the predetermined time.
192. The method of claim 189 wherein the encrypting comprises encrypting a plurality of digital assets using a single encryption key.

193. The method of claim 189 wherein the encrypting comprises encrypting a plurality of digital assets using a respective encryption key for each digital asset of the plurality of digital assets.
194. The method of claim 189 wherein the encryption is performed by a file system filter driver.
195. The method of claim 189 wherein the encryption is performed by a plug-in module.
196. A system for preventing access to an expired digital asset, the system comprising:  
  
a computing device having an application thereon that encrypts a digital asset using an encryption key, the digital asset associated with a tag specifying a time after which the digital asset should be unavailable, the encryption key having a predetermined expiration time, and destroys the encryption key after the predetermined time.
197. The system of claim 196 wherein the digital asset is marked for expiration prior to encrypting the digital asset.
198. The system of claim 196 wherein the computing device copies the digital asset from a file system of the computing device to another storage medium, copies the encryption key to the another storage medium.
199. The system of claim 196 wherein a plurality of digital assets are encrypted using a single encryption key.
200. The system of claim 196 wherein a plurality of digital assets are encrypted using a respective encryption key for each digital asset.
201. The system of claim 196 further comprising a file system filter driver in communication with the computing device to encrypt the digital asset.
202. The system of claim 196 further comprising a plug-in module in communication with the computing device to encrypt the digital asset.



203. A computer readable medium having executable instructions thereon to prevent access to an expired digital asset., the computer readable medium comprising:
- (a) instructions to encrypt a digital asset using an encryption key, the digital asset associated with a tag specifying a time after which the digital asset should be unavailable, the encryption key having a predetermined expiration time; and
  - (b) destroying the encryption key after the predetermined time.
204. The computer readable medium of claim 203 further comprising marking the digital asset for expiration prior to encrypting the digital asset.
205. The computer readable medium of claim 203 further comprising instructions to copy the encrypted digital asset from a file system to another storage medium, instructions to copy the encryption key to the another storage medium, and instructions to destroy the encryption key after the predetermined time.
206. The computer readable medium of claim 203 wherein the instructions to encrypt comprise instructions to encrypt a plurality of digital assets using a single encryption key.
207. The computer readable medium of claim 203 wherein the instructions to encrypt comprise instructions to encrypt a plurality of digital assets using a respective encryption key for each digital asset of the plurality of digital assets.
208. The computer readable medium of claim 203 wherein at least some of the instructions are performed by a file system filter driver.
209. The computer readable medium of claim 203 wherein at least some of the instructions are performed by a plug-in module.
210. A method of setting a categorization for a digital asset in a distributed computing environment, the method comprising:

- (a) receiving, from a first computing device, an asset identification tag and a first taxonomy tag having a first categorization associated with a digital asset;
- (b) receiving, from a second computing device, the asset identification tag and a second taxonomy tag having a second categorization associated with the digital asset; and
- (c) selecting one of the first taxonomy tag and the second taxonomy tag to associate with the digital asset as the active taxonomy tag.

- 211. The method of claim 210 further comprising transmitting each of the first taxonomy tag and the second taxonomy tag to each of the first computing device and the second computing device.
- 212. The method of claim 210 wherein selecting comprises selecting the taxonomy tag that allows the fewest end-users access to the digital asset.
- 213. The method of claim 210 wherein selecting comprises selecting the taxonomy tag that allows the largest end-users access to the digital asset.
- 214. The method of claim 210 wherein selecting comprises selecting the taxonomy tag that allows the fewest actions to be performed on the digital asset.
- 215. The method of claim 210 wherein selecting comprises selecting the taxonomy tag that allows the most actions to be performed on the digital asset.
- 216. The method of claim 210 wherein selecting comprises selecting the taxonomy tag that has the earliest creation date.
- 217. The method of claim 210 further comprising transmitting the second taxonomy tag to the first computing device and transmitting the first taxonomy tag to the second computing device.
- 218. The method of claim 210 wherein steps (a), (b), and (c) are performed by a server of a distributed computing environment.
- 219. The method of claim 210 wherein steps (a), (b), and (c) are performed by a client of a distributed computing environment.



220. A system for setting a categorization for a digital asset in a distributed computing environment, the system comprising:
- (a) a first computing device transmitting via a network, an asset identification tag and a first taxonomy tag having a first categorization associated with a digital asset;
  - (b) a second computing device transmitting via the network, the asset identification tag and a second taxonomy tag having a second categorization associated with the digital asset; and
  - (c) another computing device in communication with the network that receives the first taxonomy tag and the second taxonomy tag and selects one of the first taxonomy tag and the second taxonomy tag as the active taxonomy tag.
221. The system of claim 220 wherein the another computing device comprises the first computing device.
222. The system of claim 220 wherein the another computing device comprises the second computing device.
223. The system of claim 220 wherein the another computing device transmits each of the first taxonomy tag and the second taxonomy tag to each of the first computing device and the second computing device.
224. The system of claim 220 wherein the another computing device selects the taxonomy tag that allows the fewest end-users access to the digital asset.
225. The system of claim 220 wherein the another computing device selects the taxonomy tag that allows the largest end-users access to the digital asset.
226. The system of claim 220 wherein the another computing device selects the taxonomy tag that allows the fewest actions to be performed on the digital asset.
227. The system of claim 220 wherein the another computing device selects taxonomy tag that allows the most actions to be performed on the digital asset.

228. The system of claim 220 wherein the another computing device selects the taxonomy tag that has the earliest creation date.
229. The system of claim 220 wherein the another computing device transmitting the second taxonomy tag to the first computing device and transmitting the first taxonomy tag to the second computing device.
230. The system of claim 220 wherein the another computing device is a server of a distributed computing environment.
231. The system of claim 220 wherein the another computing device is a client of a distributed computing environment.
232. A computer readable medium having executable instructions thereon to set a categorization for a digital asset in a distributed computing environment, the computer readable medium comprising:
- (a) instructions to receive, from a first computing device, an asset identification tag and a first taxonomy tag having a first categorization associated with a digital asset;
  - (b) instructions to receive, from a second computing device, the asset identification tag and a second taxonomy tag having a second categorization associated with the digital asset; and
  - (c) instruction to select one of the first taxonomy tag and the second taxonomy tag to associate with the digital asset as the active taxonomy tag.
233. The computer readable medium of claim 232 wherein the instruction to select comprises instructions to select the taxonomy tag that allows the fewest end-users access to the digital asset.
234. The computer readable medium of claim 232 wherein the instruction to select comprises instructions to select the taxonomy tag that allows the largest end-users access to the digital asset.
235. The computer readable medium of claim 232 wherein the instruction to select comprises instructions to select the taxonomy tag that allows the fewest actions to be performed on the digital asset.



236. The computer readable medium of claim 232 wherein the instruction to select comprises instructions to select the taxonomy tag that allows the most actions to be performed on the digital asset.
237. The computer readable medium of claim 232 wherein the instruction to select comprises instructions to select the taxonomy tag that has the earliest creation date.
238. The computer readable medium of claim 232 further comprising instructions to transmit the second taxonomy tag to the first computing device and transmitting the first taxonomy tag to the second computing device.
239. The computer readable medium of claim 232 wherein the instruction are executed by a server of a distributed computing environment.
240. The computer readable medium of claim 232 wherein the instruction are executed by a client of a distributed computing environment.
241. A method of propagating categorization information associated with a digital asset between clients in a distributed computing environment the environment having a central computing device that stores an asset identification tag for at least one digital asset and one or more clients that locally stores respective digital assets, the method comprising:
- (a) generating, by a first client, a meta-data set including categorization information associated with the digital asset;
  - (b) associating the meta-data set with the digital asset such that the meta-data has at least one characteristic of the digital asset; and
  - (c) transferring the digital asset and meta-data set to a second client of the distributed computing system.
242. The method of claim 241 further comprising accessing the meta-data, by the second client, prior to accessing to the digital asset.
243. The method of claim 241 wherein the set is set transferred in an e-mail header.

244. The method of claim 243 wherein the set is transferred in a comments field of the e-mail header.
245. The method of claim 243 wherein the set is transferred in a keywords field of the e-mail header.
246. The method of claim 243 wherein the set is transferred in an optional-field field of the e-mail header.
247. The method of claim 241 wherein the set contains a hash value.
248. A system for propagating categorization information associated with a digital asset between clients in a distributed computing environment the environment having a central computing device that stores an asset identification tag for at least one digital asset and one or more clients that locally stores respective digital assets, the system comprising:
- (a) a first client in communication with a network; and
  - (b) a second client in communication with a network, the second client generates a meta-data set including categorization information associated with the digital asset, associates the meta-data set with the digital asset such that the meta-data has at least one characteristic of the digital asset, and transfers the digital asset and meta-data set to the first client of the distributed computing system.
249. The system of claim 248 wherein the first client accesses the meta-data prior to accessing to the digital asset.
250. The system of claim 248 wherein the set is set transferred from the second client to the first client via an e-mail header.
251. The system of claim 250 wherein the set is transferred in a comments field of the e-mail header.
252. The system of claim 250 wherein the set is transferred in a keywords field of the e-mail header.
253. The system of claim 250 wherein the set is transferred in an optional-field field of the e-mail header.



254. The system of claim 248 wherein the set contains a hash value.
255. A computer readable medium having executable instructions thereon to propagate categorization information associated with a digital asset between clients in a distributed computing environment the environment having a central computing device that stores an asset identification tag for at least one digital asset and one or more clients that locally stores respective digital assets, the computer readable medium comprising:
- (a) instructions to generate, by a first client, a meta-data set including categorization information associated with the digital asset;
  - (b) instructions to associate the meta-data set with the digital asset such that the meta-data has at least one characteristic of the digital asset; and
  - (c) instruction to transferring the digital asset and meta-data set to a second client of the distributed computing system.
256. The computer readable medium of claim 255 further comprising instructions to access the meta-data, by the second client, prior to accessing to the digital asset.
257. The computer readable medium of claim 255 further comprising instructions to transfer the meta-data set in an e-mail header.
258. The computer readable medium of claim 257 further comprising instructions to transfer the meta-data set in a comments field of the e-mail header.
259. The computer readable medium of claim 257 further comprising instructions to transfer the meta-data set in a keywords field of the e-mail header.
260. The computer readable medium of claim 257 further comprising instructions to transfer the meta-data set in an optional-field field of the e-mail header.
261. The computer readable medium of claim 255 wherein the meta-data set contains a hash value.

262. A method of classifying digital assets in a computing system, the method comprising:
- (a) creating, by a client, a digital asset;
  - (b) presenting at least one tag of an adaptive taxonomy to an end-user of the client, the at least one tag describing a classification of the digital asset and having a first associated policy that is applied to the digital asset when the at least one tag is associated with the digital asset; and
  - (c) providing the end-user with a graphical user interface when the end-user performs a specified action, the graphical user interface allowing the end-user to create an alias for the at least one tag, the alias associated with the tag and having a second policy.
263. The method of claim 262 wherein the first policy and the second policy are substantially similar.
264. The method of claim 262 wherein the first policy and the second policy are different.
265. The method of claim 262 wherein the presenting occurs during the creation of the digital asset.
266. The method of claim 262 wherein the presenting occurs after the creation of the digital asset.
267. The method of claim 262 further comprising transmitting the alias to a central location for review by an administrator.
268. The method of claim 267 further comprising accepting the alias by the administrator thereby making the alias accessible to a second end-user.
269. The method of claim 267 further comprising rejecting the alias by the administrator thereby making the alias unavailable to a second end-user.
270. The method of claim 262 further comprising checking for the existence of the alias.
271. The method of claim 262 further comprising presenting an existing alias along with the tag to end-user.



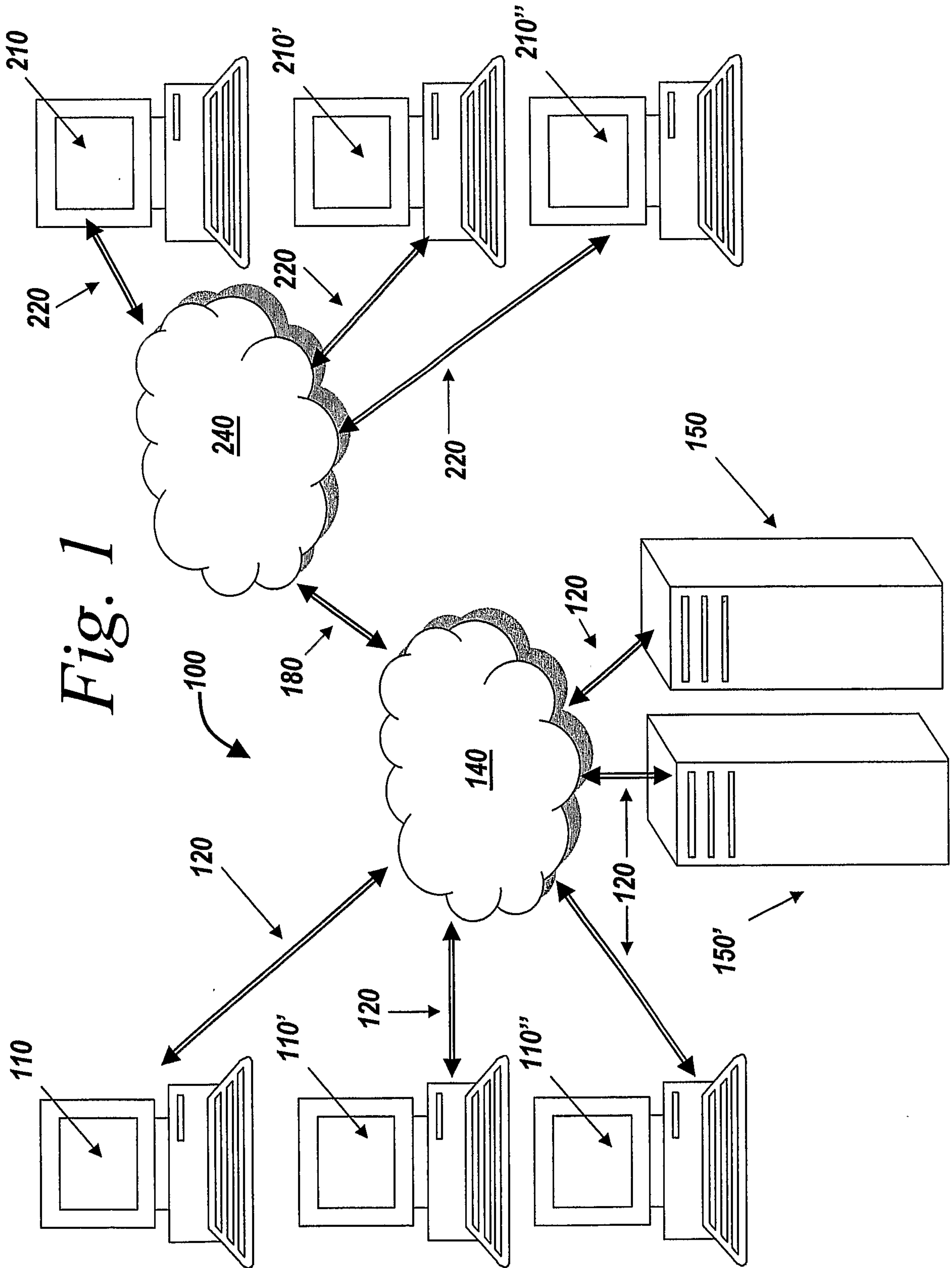
272. A system for classifying digital assets in a computing system, the system comprising:
- (a) a server connected to a network; and
  - (b) a client in communication with the server, the client creating a digital asset, presenting at least one tag of an adaptive taxonomy to an end-user of the client, the at least one tag describing a classification of the digital asset and having a first associated policy that is applied to the digital asset when the at least one tag is associated with the digital asset, and providing the end-user with a graphical user interface when the end-user performs a specified action, the graphical user interface allowing the end-user to create an alias for the at least one tag, the alias associated with the tag and having a second policy.
273. The system of claim 272 wherein the first policy and the second policy are substantially similar.
274. The system of claim 272 wherein the first policy and the second policy are different.
275. The system of claim 272 wherein the client presents the graphical user interface during the creation of the digital asset.
276. The system of claim 272 wherein the client presents the graphical user interface after the creation of the digital asset.
277. The system of claim 272 further wherein the client transmits the alias to the server for review by an administrator.
278. The system of claim 277 wherein the administrator accepts the alias at the server thereby making the alias accessible to a second end-user.
279. The system of claim 277 wherein the administrator rejects the alias at the server thereby making the alias unavailable to a second end-user.
280. The system of claim 272 wherein the client or the server checks for the existence of the alias.

281. The system of claim 272 wherein the client presents an existing alias along with the tag to end-user.
282. A computer readable medium having executable instructions thereon to classify digital assets in a computing system, the computer readable medium comprising:
- (a) instructions to create, by a client, a digital asset;
  - (b) instructions to present at least one tag of an adaptive taxonomy to an end-user of the client, the at least one tag describing a classification of the digital asset and having a first associated policy that is applied to the digital asset when the at least one tag is associated with the digital asset; and
  - (c) instruction to provide the end-user with a graphical user interface when the end-user performs a specified action, the graphical user interface allowing the end-user to create an alias for the at least one tag, the alias associated with the tag and having a second policy.
283. The computer readable medium of claim 282 wherein the first policy and the second policy are substantially similar.
284. The computer readable medium of claim 282 wherein the first policy and the second policy are different.
285. The computer readable medium of claim 282 wherein the instructions to present comprise instructions to present during the creation of the digital asset.
286. The computer readable medium of claim 282 wherein the instructions to present comprise instructions to present after the creation of the digital asset.
287. The computer readable medium of claim 282 further comprising instructions to transmit the alias to a central location for review by an administrator.



288. The computer readable medium of claim 287 further comprising instructions to make the alias accessible to a second end-user when the administrator accepts the alias.
289. The computer readable medium of claim 287 further comprising instructions to make the alias not accessible to a second end-user when the administrator rejects the alias.
290. The computer readable medium of claim 282 further comprising instructions to check for the existence of the alias.
291. The computer readable medium of claim 282 further comprising instructions to present an existing alias along with the tag to end-user.

1/11





2/11

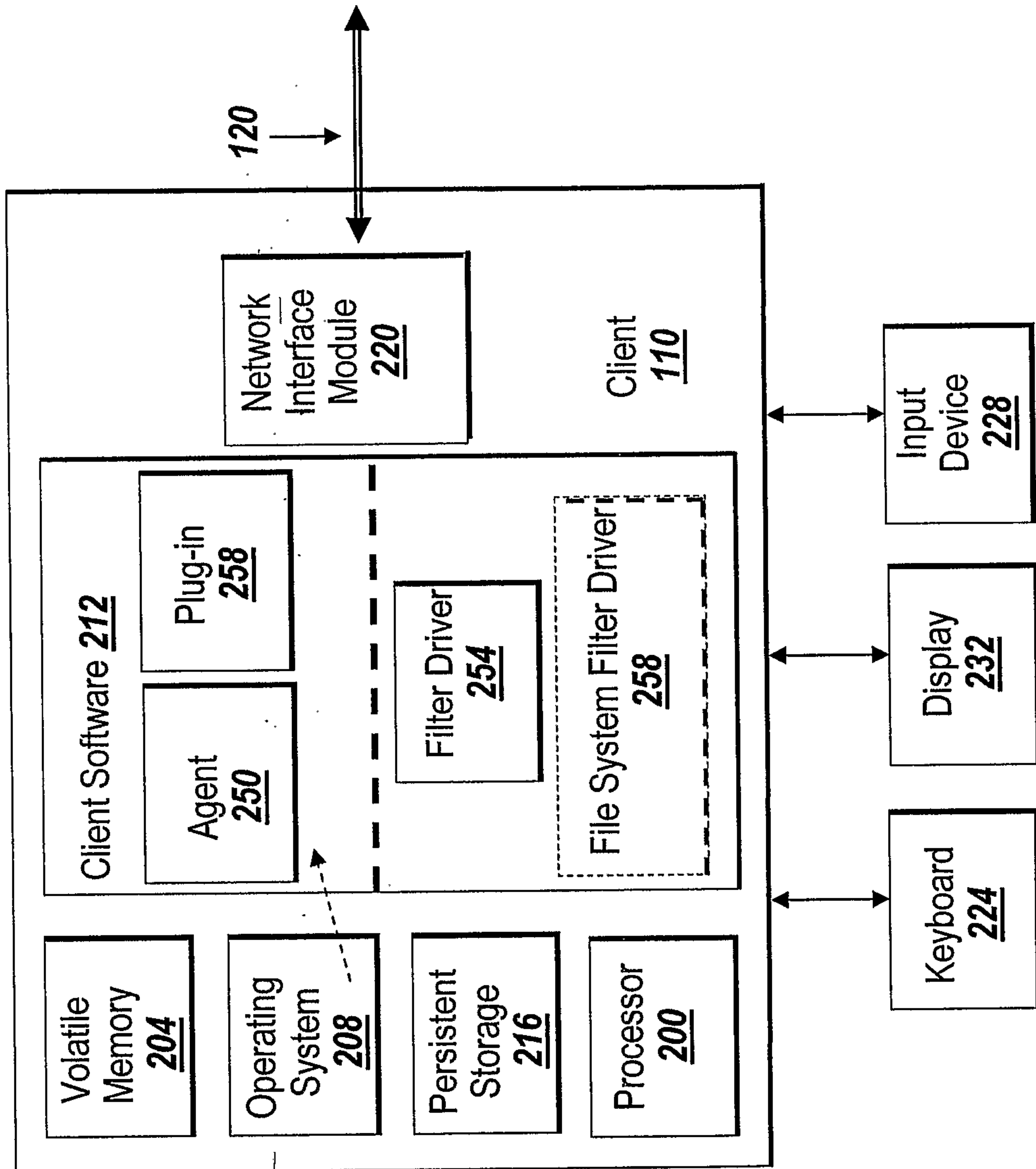
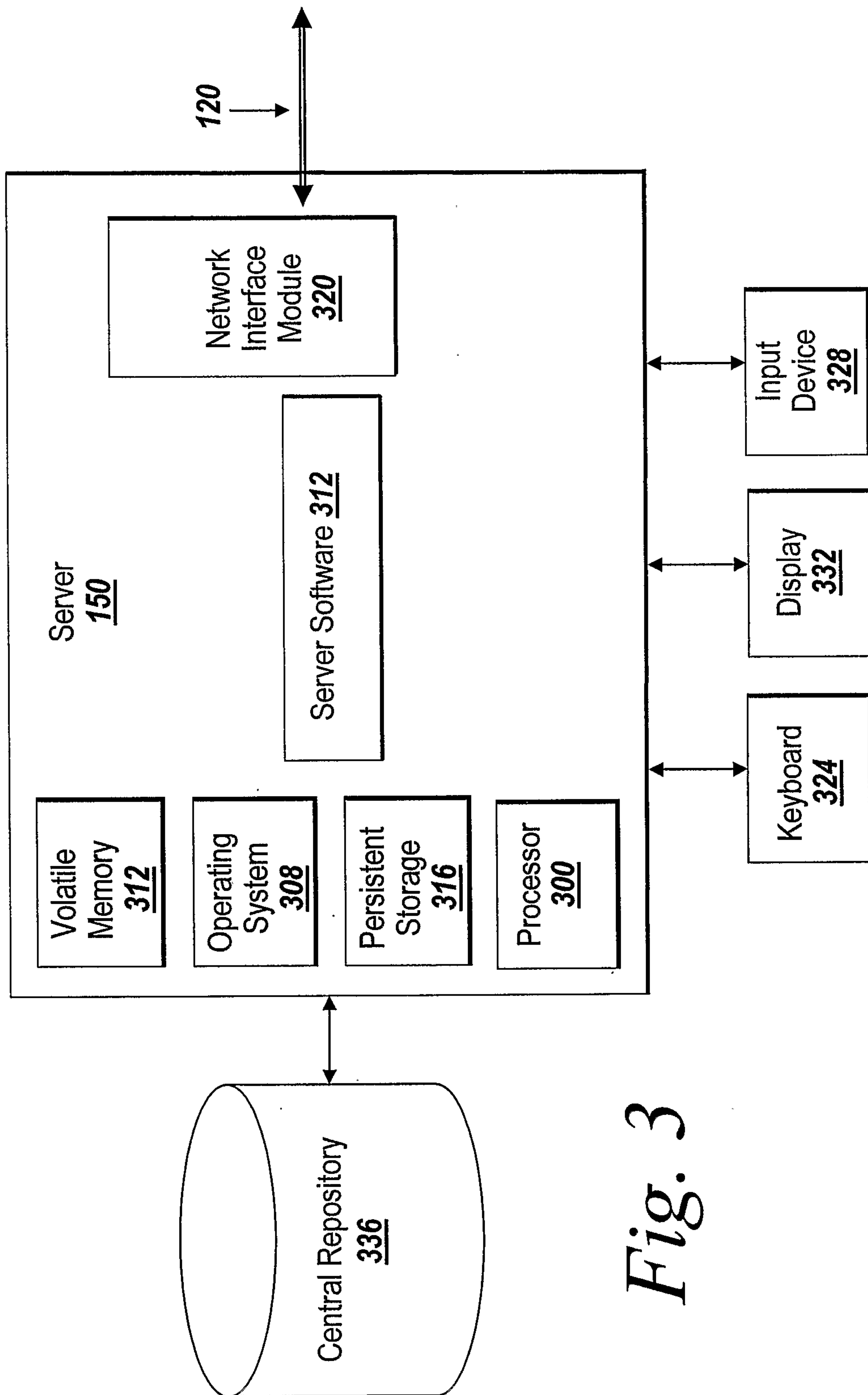


Fig. 2

3/11

*Fig. 3*



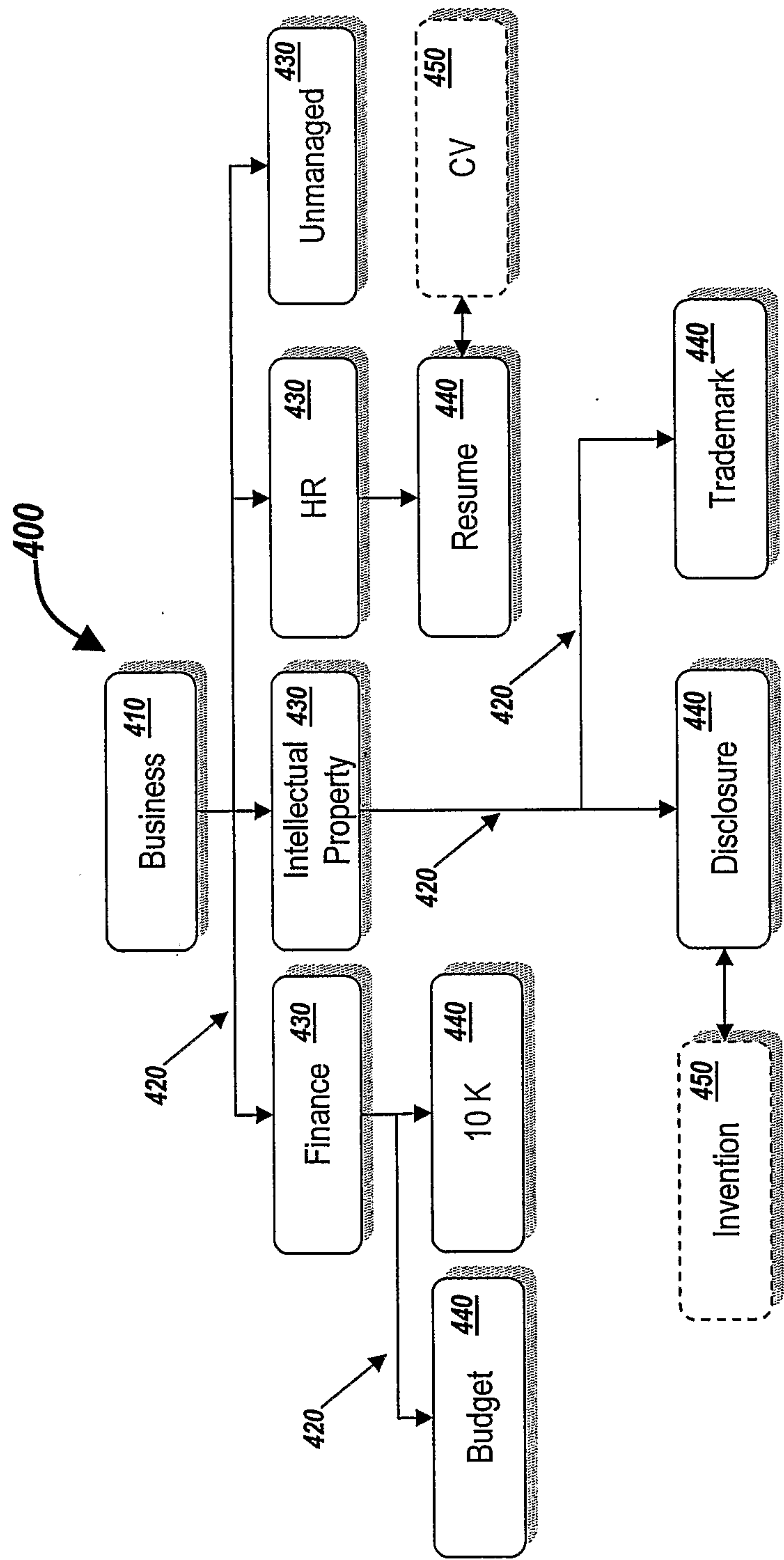
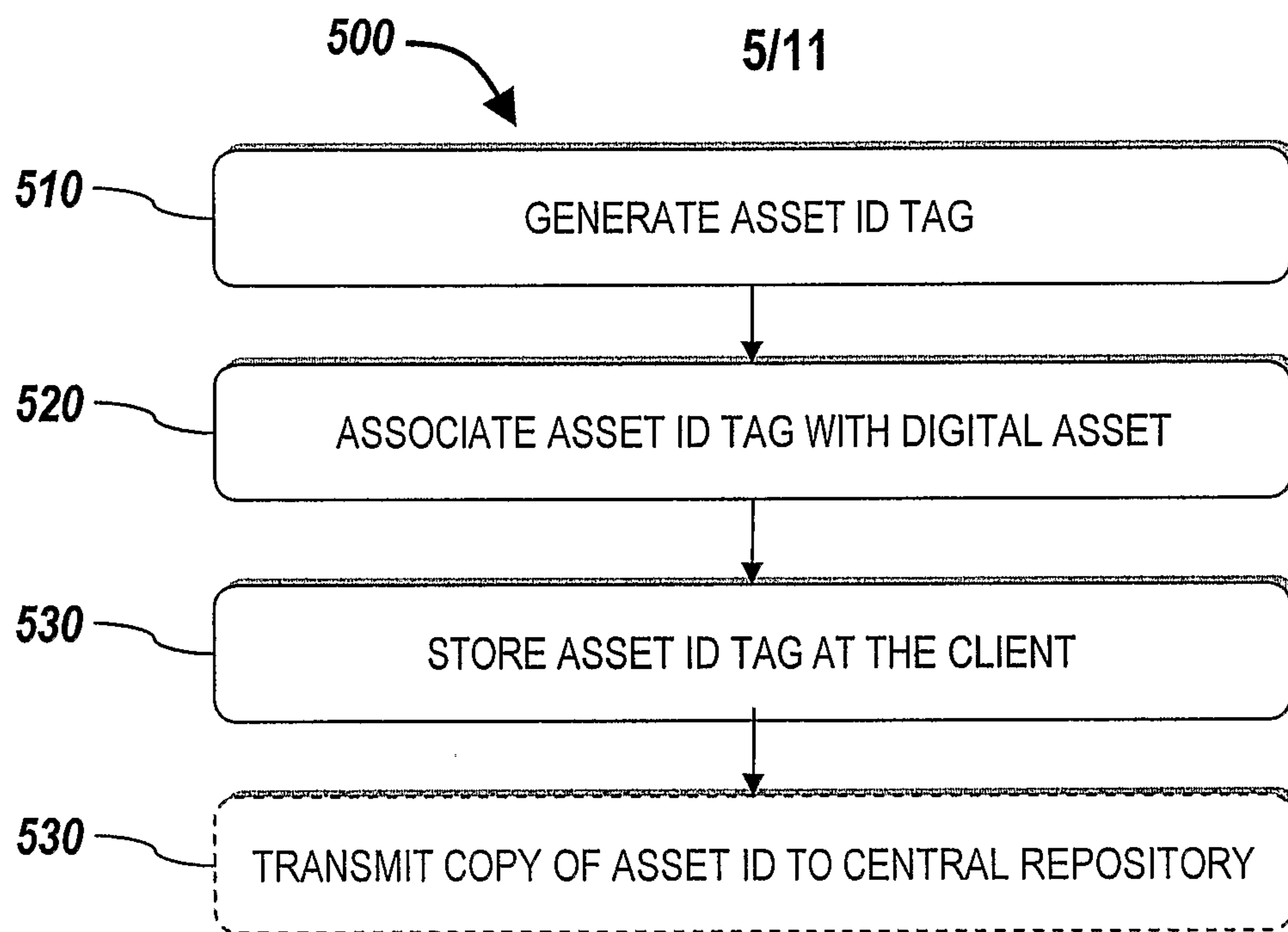
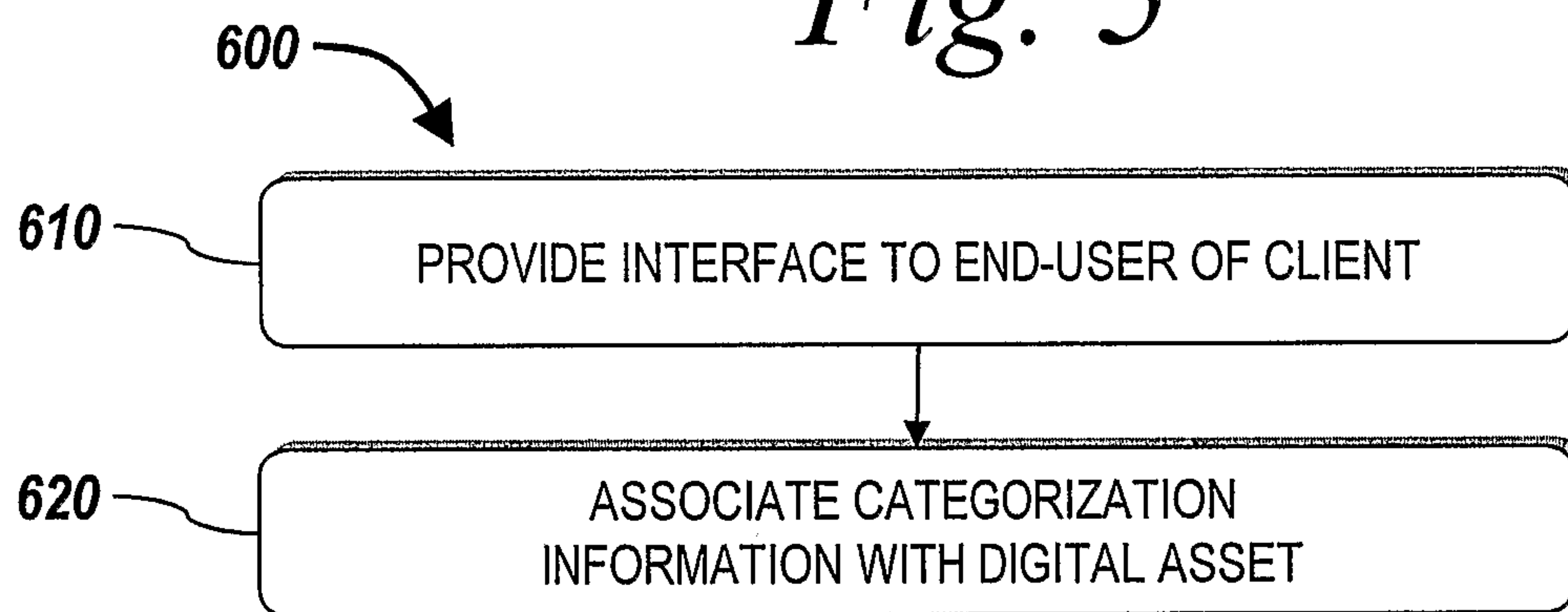
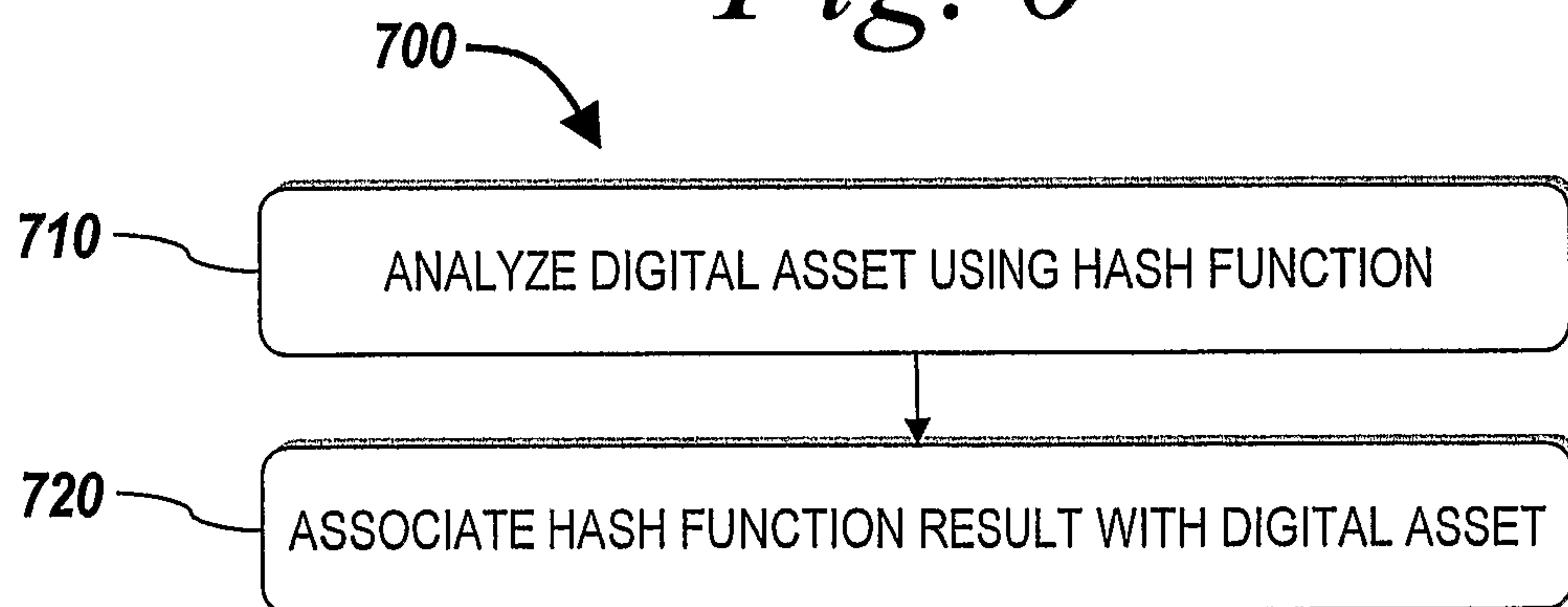
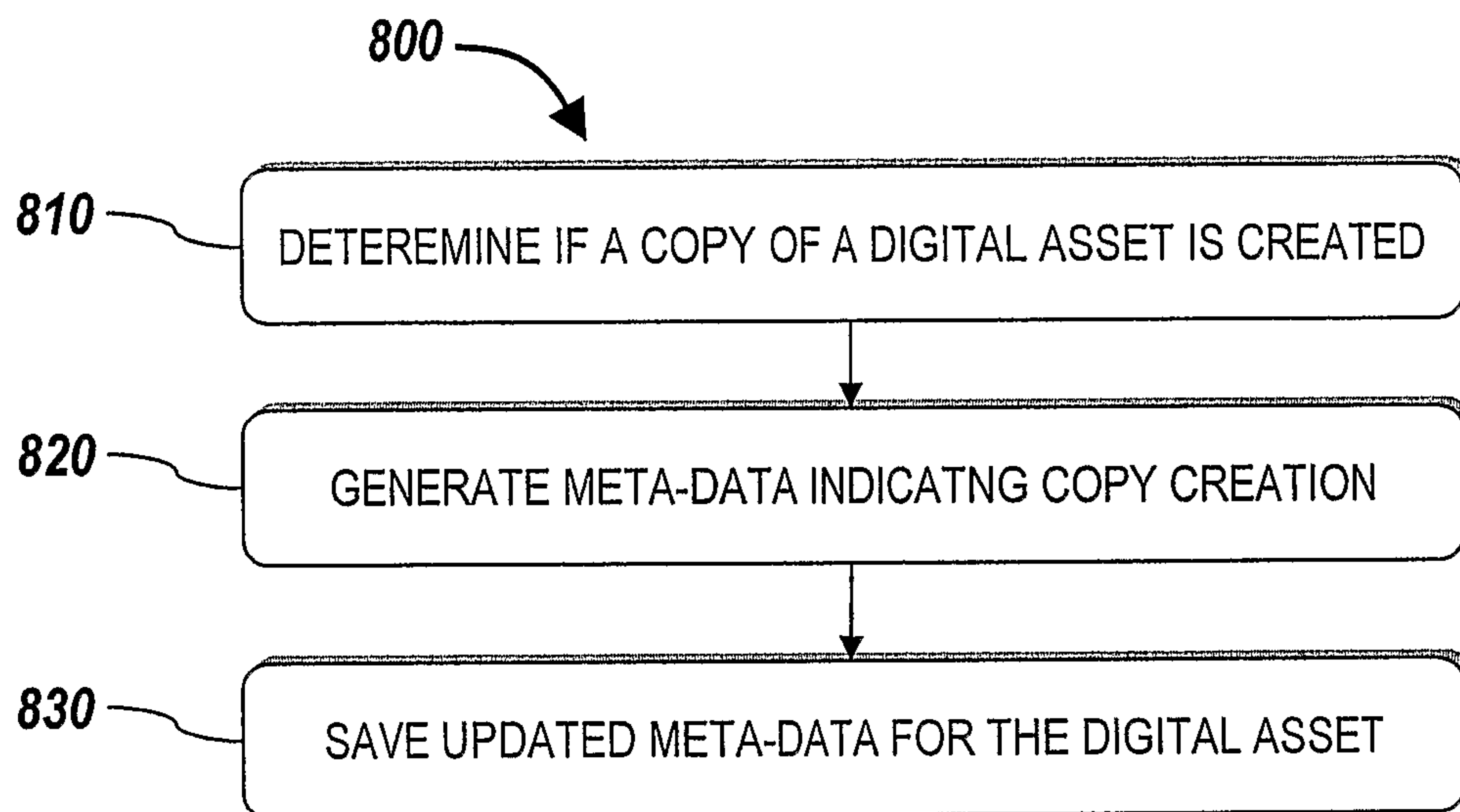
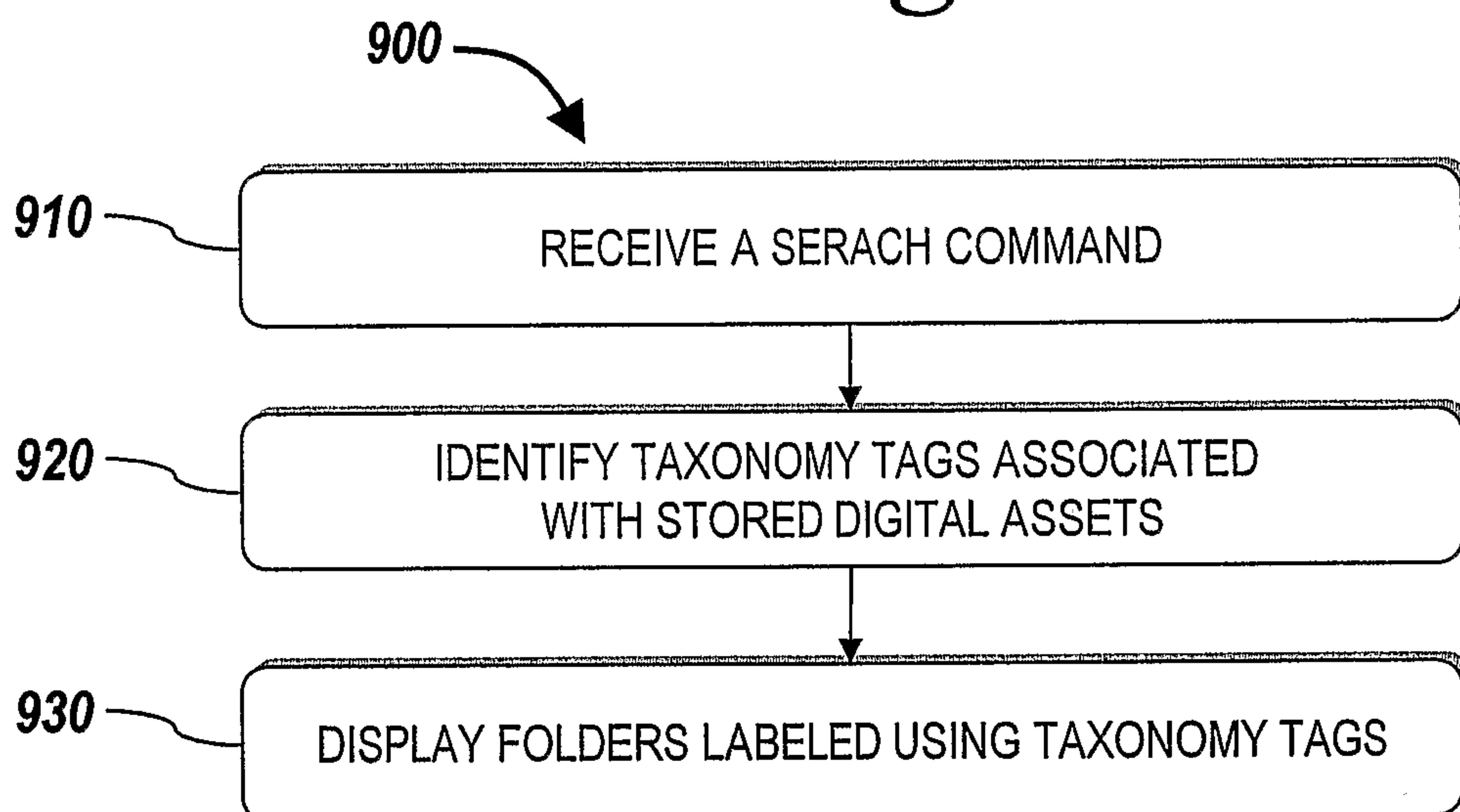


Fig. 4

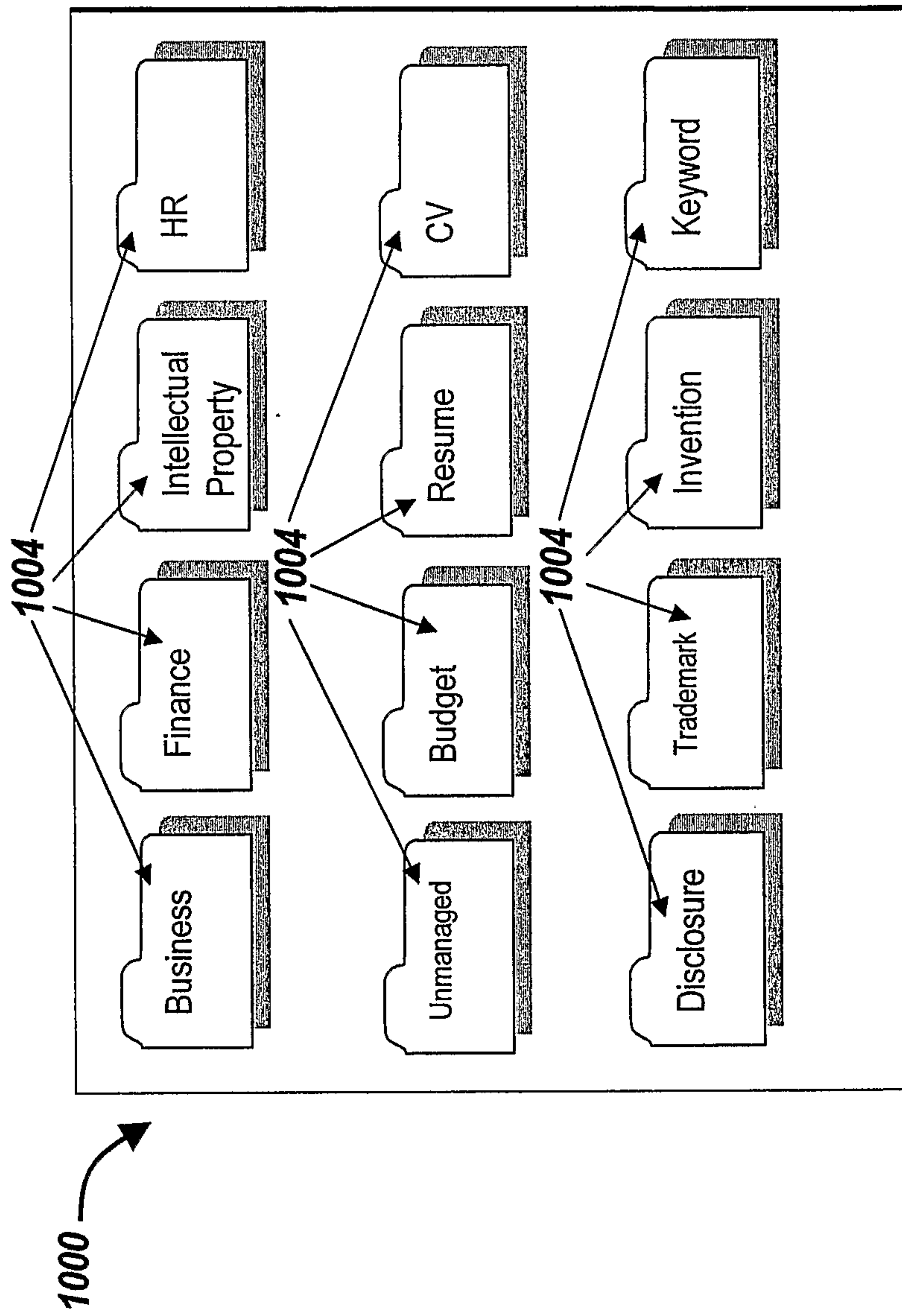
*Fig. 5**Fig. 6**Fig. 7*



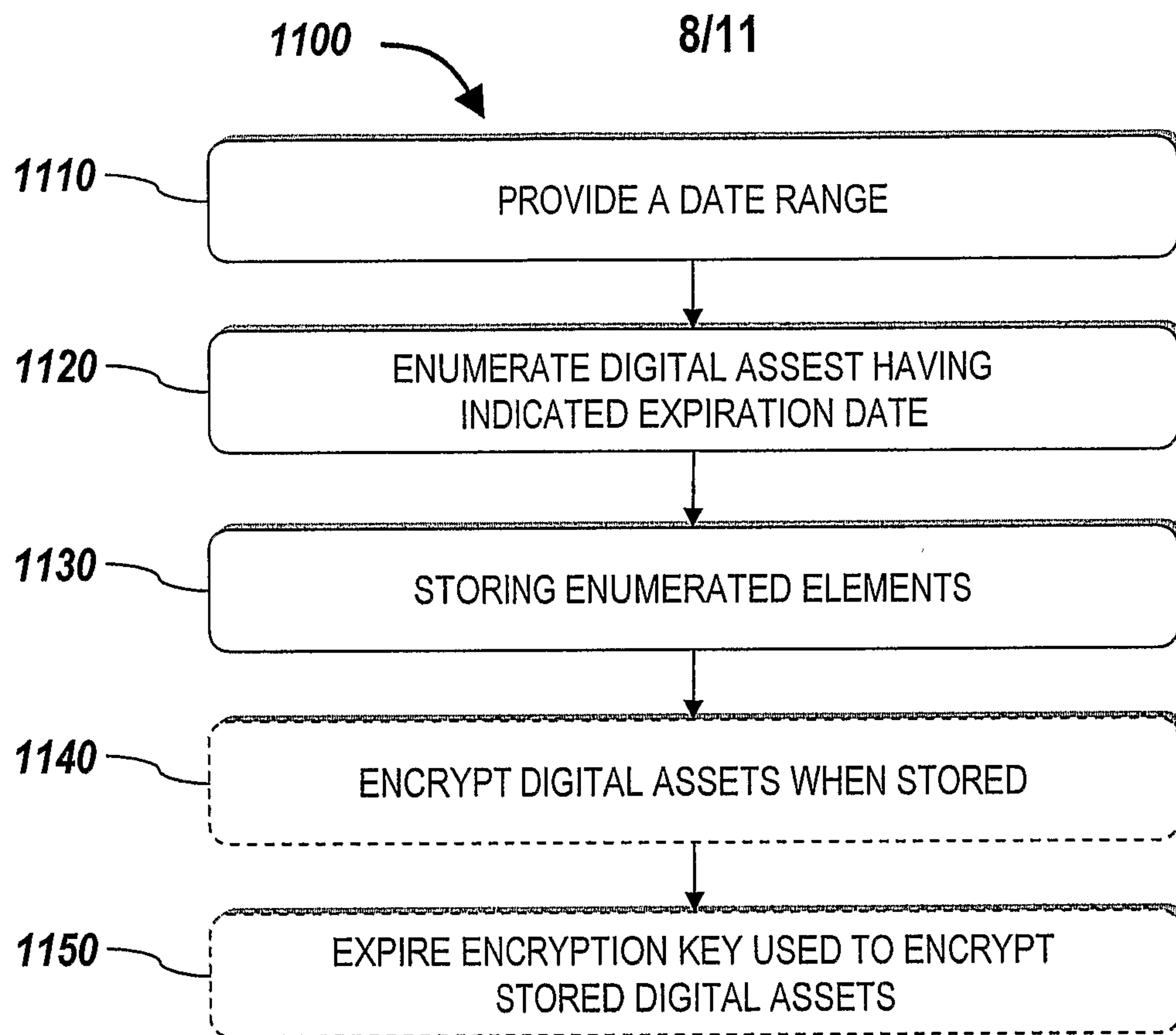
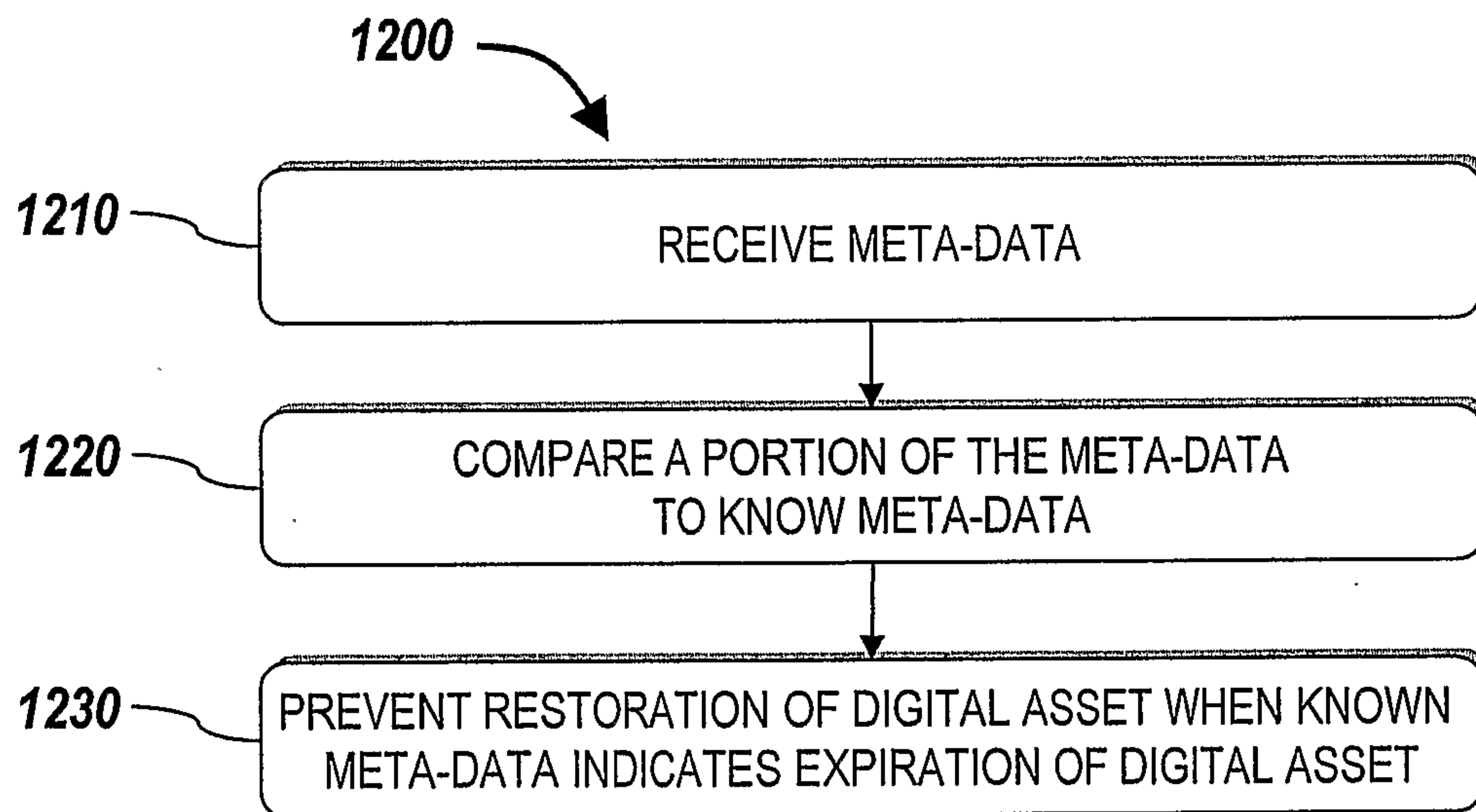
6/11

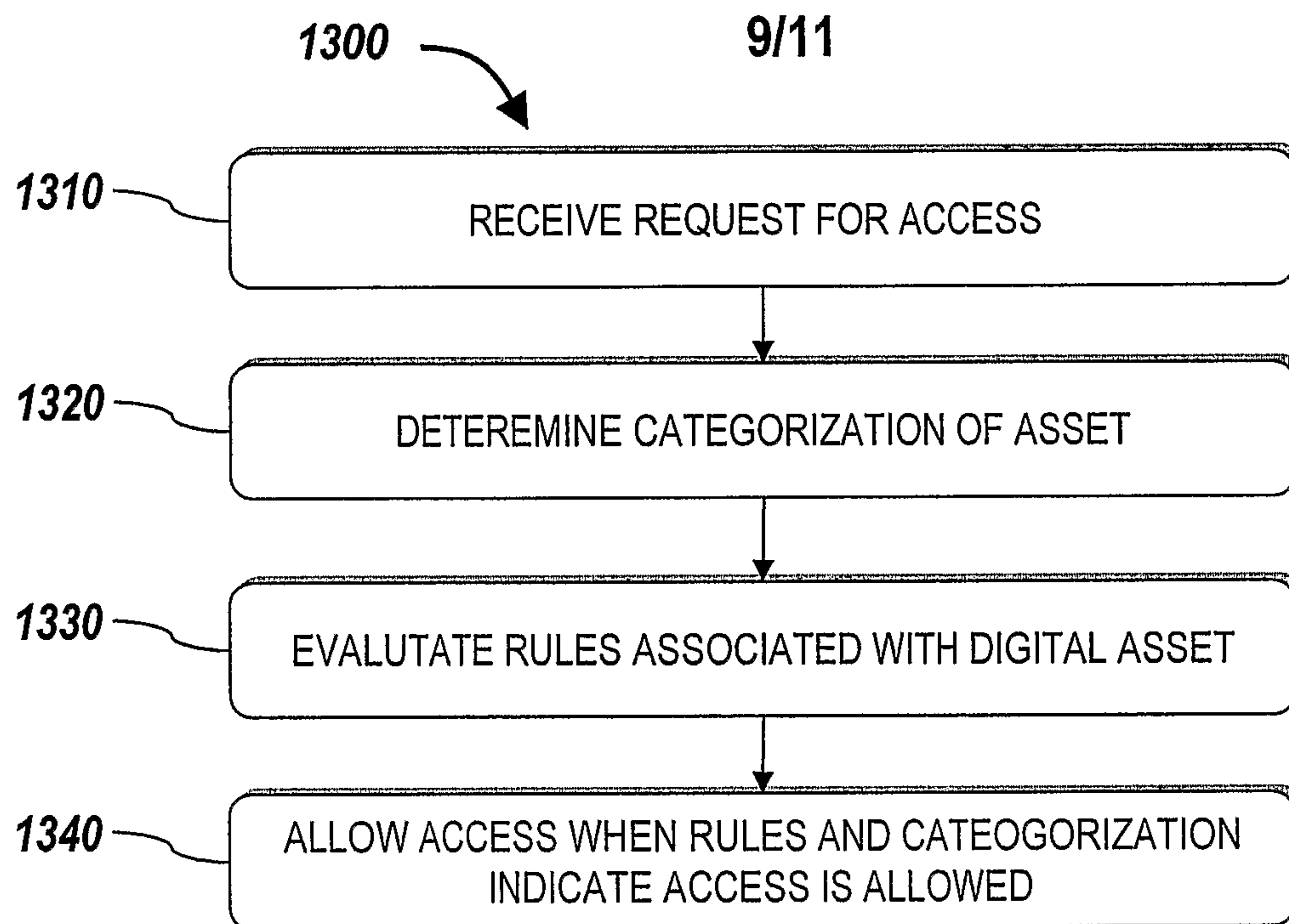
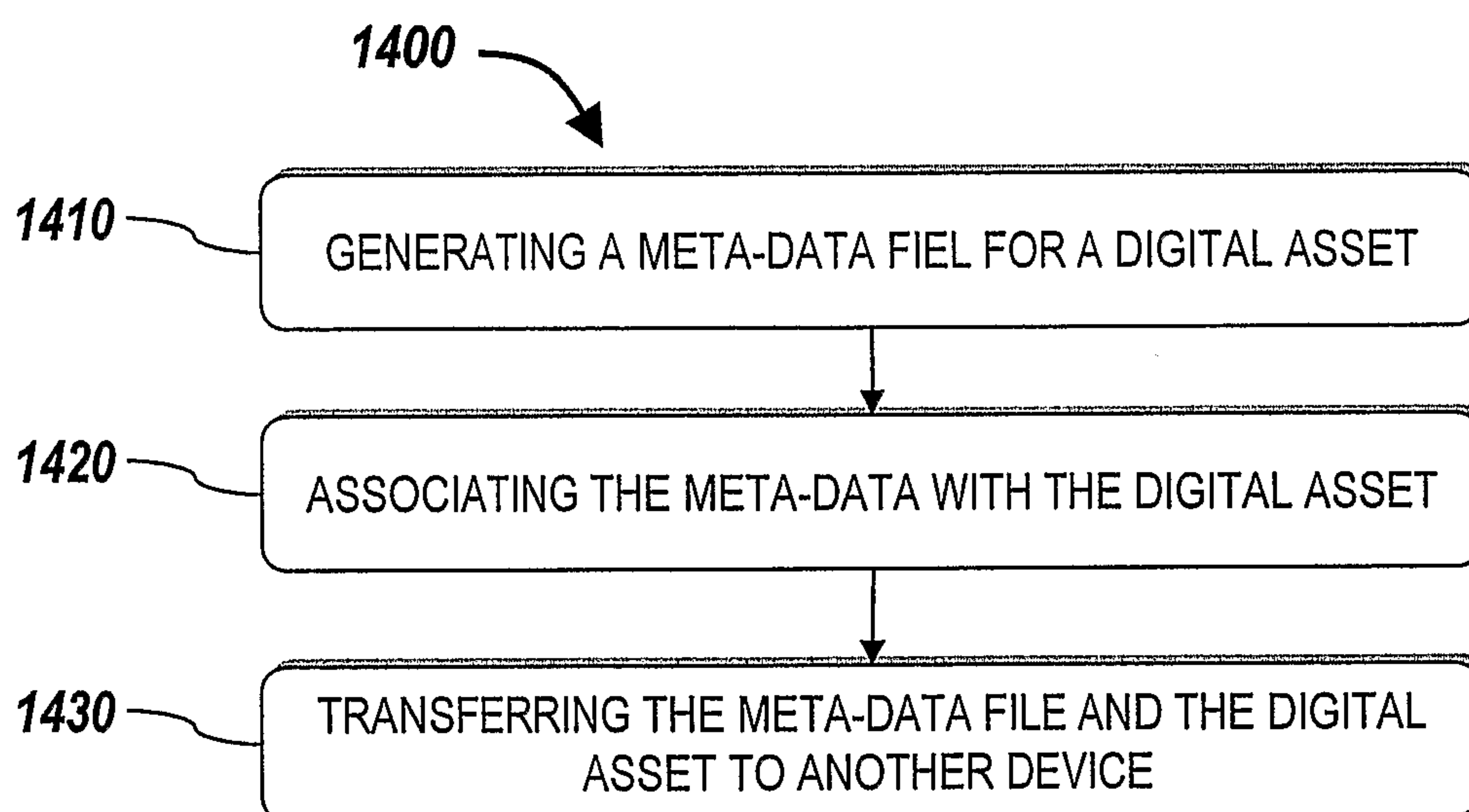
*Fig. 8**Fig. 9*

7/11

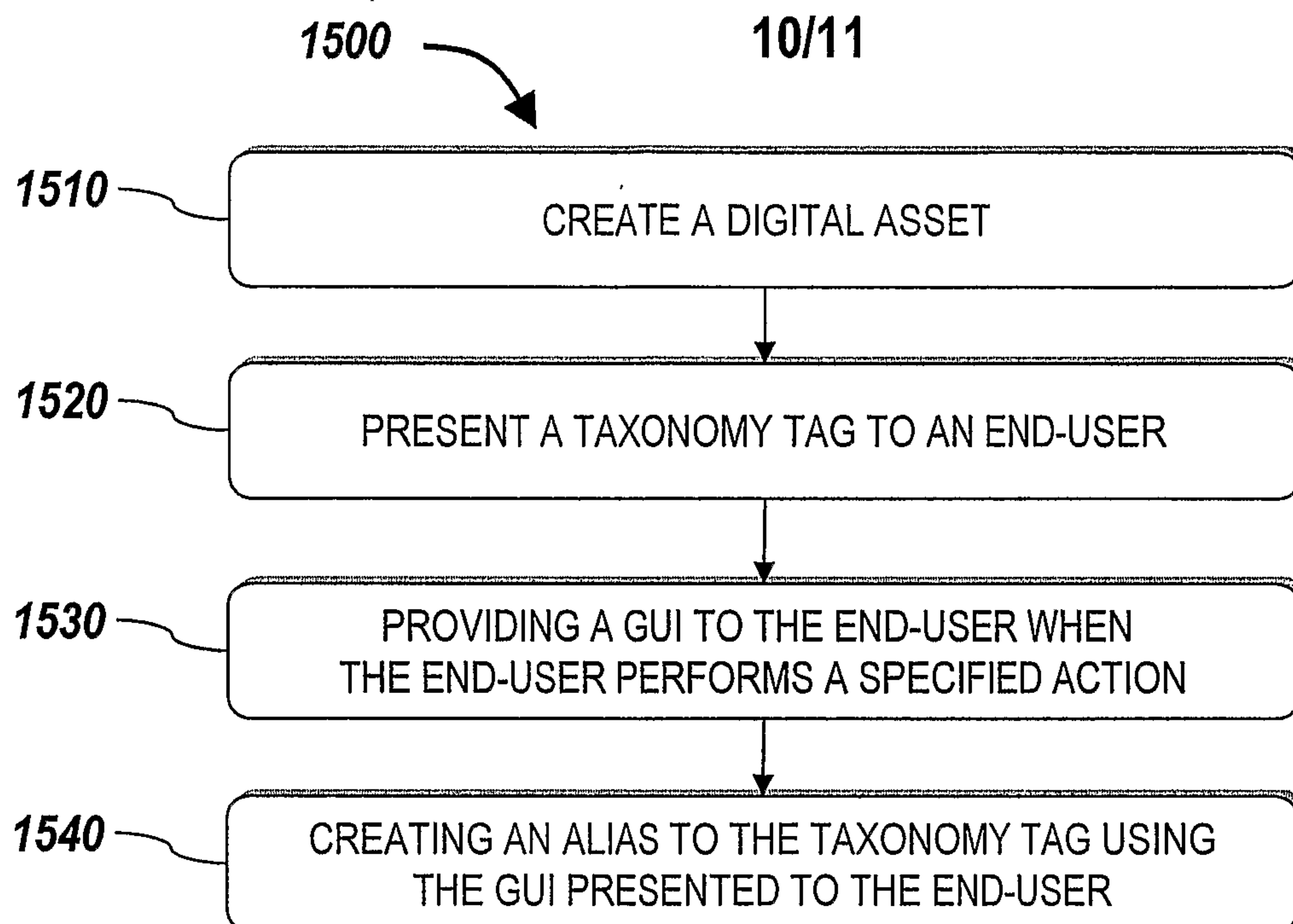
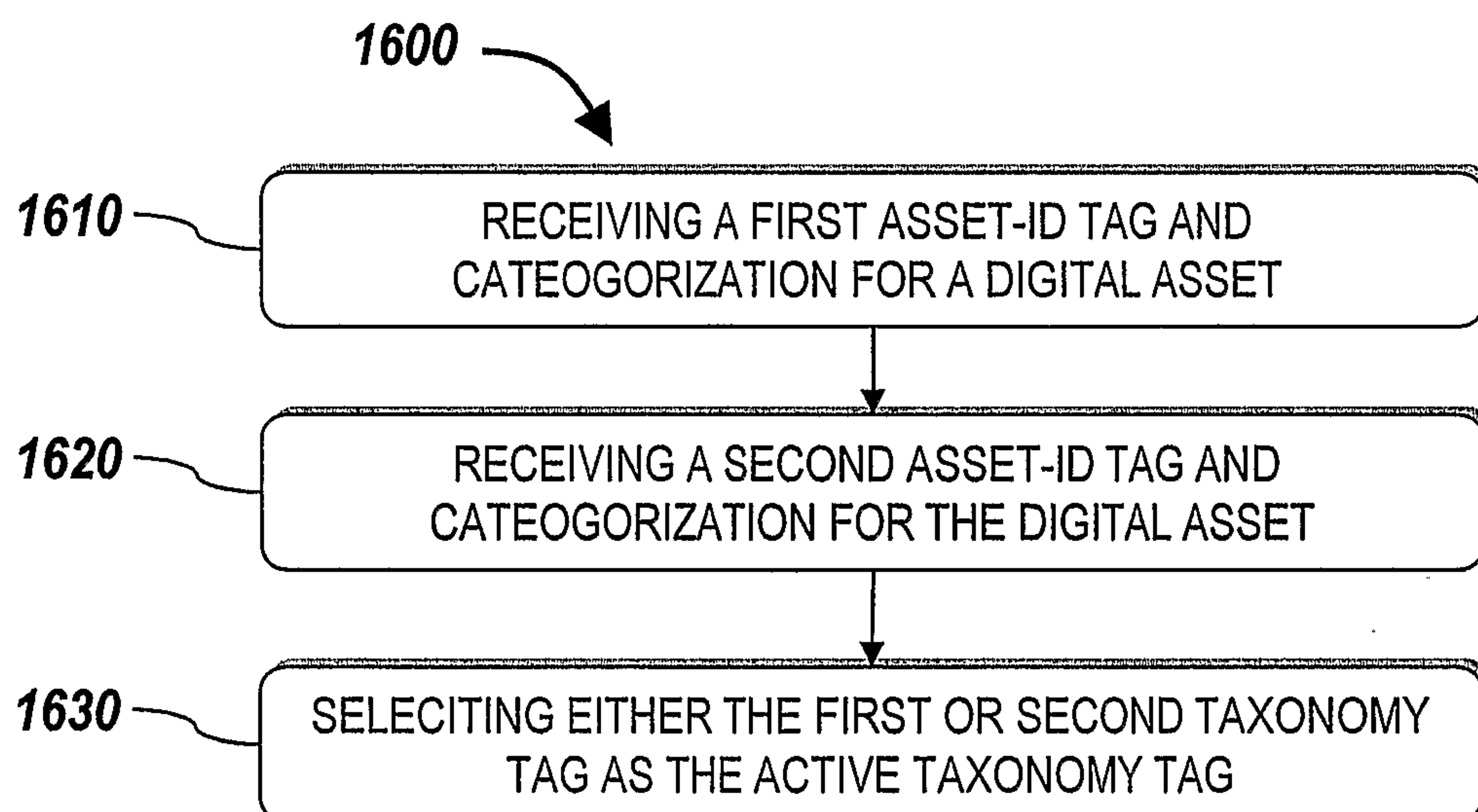
*Fig. 10*



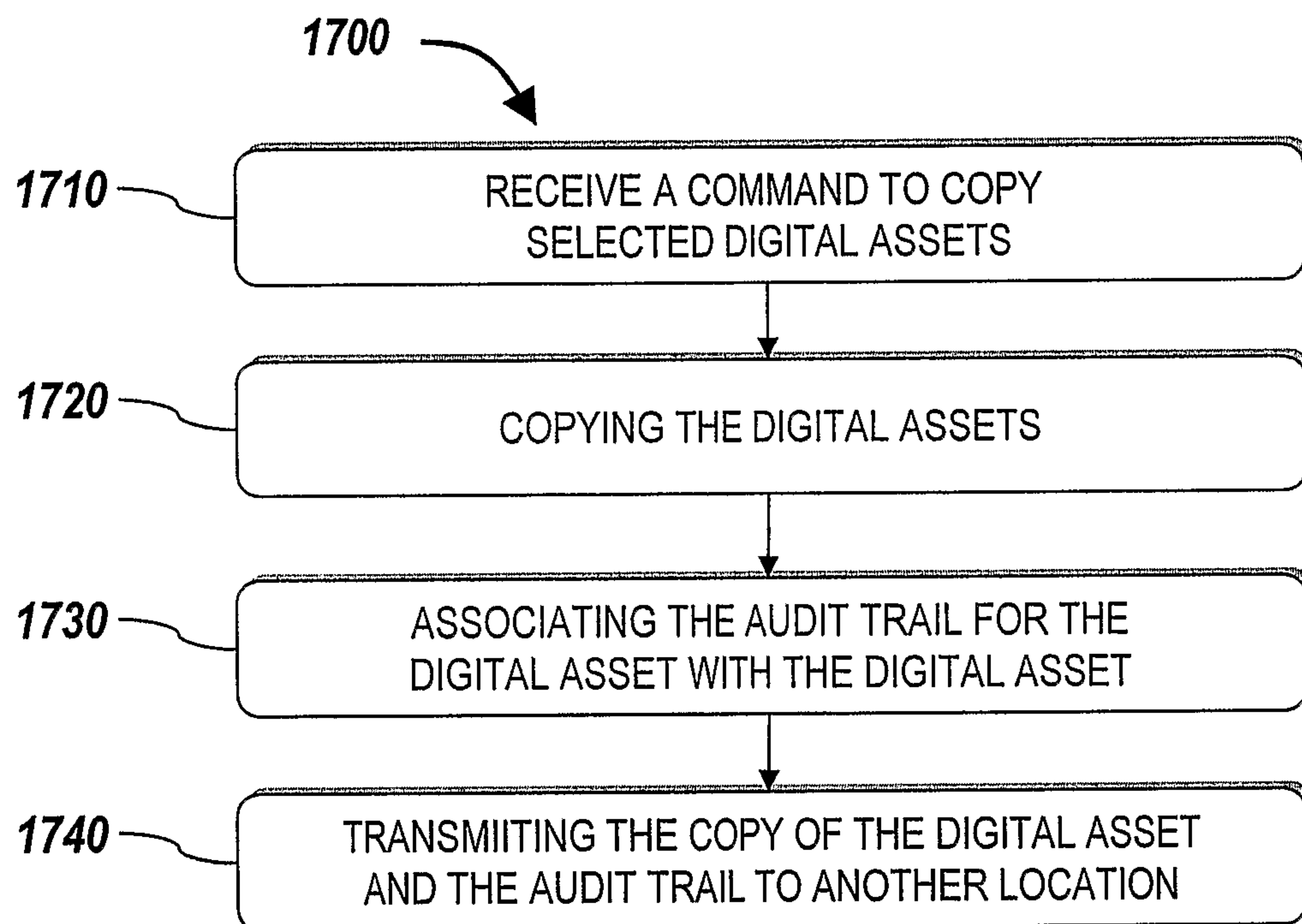
*Fig. 11**Fig. 12*

*Fig. 13**Fig. 14*



*Fig. 15**Fig. 16*

11/11

*Fig. 17*



